Department of Computer Science and Engineering

Vision

Excellence in Computer Science and Engineering education and research.

Mission

1. Strive for academic excellence in Computer Science and Engineering through a creative teaching learning process.
2. Transform students into technically competent, socially responsible and ethical Computer Science professionals.
3. Create Centres of Excellence in leading areas of Computer Science and Engineering.
4. Incubate, apply and spread innovative ideas by collaborating with relevant industries and R&D labs through focused research groups.
5. Attain these through continuous team work by a group of committed faculty, transforming the Computer Science and Engineering department as a leader in imparting Computer Science and Engineering education and research.

Programme Educational Objectives for B.E CSE

PEO1: Graduates will be able to perform in technical/managerial roles ranging from design, development, problem solving to production support in software industries and R&D sectors.

PEO2: Graduates will be able to successfully pursue higher education in reputed institutions.

PEO3: Graduates will have the ability to adapt, contribute and innovate new technologies and systems in the key domains of Computer Science and Engineering.

PEO4: Graduates will be ethically and socially responsible solution providers and entrepreneurs in Computer Science and other engineering disciplines.

Programme Outcomes for B.E CSE

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs) for B.E(CSE) Programme

PSO1: (Cognitive Outcome)

Ability to solve complex Knowledge Engineering problems by building systems across domains including Systems Engineering, Software Development & Engineering, Networks & Security, Data Mining and Artificial Intelligence.

PSO2: (Skill Outcome)

Ability to apply technical and research based skills learnt through professional society events, certification programs, projects and lab exercises to provide sustainable solutions to Computer Science and Engineering problems related to the society and environment.

PSO3: (Attitudinal and Behavioral Outcome)

Ability to practice as an ethical Software Engineer and/or Researcher in the evolving disciplines of Computer Science and Engineering and its allied application domains by employing soft and project management skills learnt through internships, project work and/or collaborative projects with industry.

PEO Vs Mission mapping

Mission/PEO	1	2	3	4
1				
2				
3				
4				
5				

Strong Correlation

Medium Correlation

Low Correlation

PO Vs PEO mapping

PEO/PO	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												
4												

Graduate Attributes

1. Engineering knowledge
2. Problem analysis
3. Design & Development of solutions
4. Investigation of Complex Problem
5. Modern tool usage
6. Engineer and society
7. Environment& sustainability
8. Ethics
9. Individual & team work
10. Communication
11. Project management & finance
12. Life-long learning

PO Vs Graduate Attributes

PO\GA	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses*

Semester	Theory						Theory cum Practical	Practical		Special Courses	Credits
	1	2	3	4	5	6	7	8	9	10	
I	**14MA110** **Engineering Mathematics - I** **(3)**	**14PH120** **Physics** **(3)**	**14CH130** **Chemistry** **(3)**	**14EG140** **English** **(3)**	**14ES150** **Basics of Civil and Mechanical Engineering** **(2)**	**14ES160** **Basics of Electrical and Electronics Engineering** **(2)**	**14ME170** **Engineering Graphics** **(3)**	**14PH180** **Physics Lab** **(1)**	**14CH190** **Chemistry Lab** **(1)**	**-**	**21**
II	**14CS210** **Engineering Mathematics - II** **(3)**	**14CS220** **Electronic Devices and Circuits** **(3)**	**14CS230** **Digital Circuits** **(3)**	**14CS241** **Computer Organization and Microprocessors** **(3)**	**14CS250** **Environment Science** **(3)**	**--**	**14CS270** **Problem Solving using Computers** **(3)**	**14CS280** **Digital Circuits Lab** **(1)**	**14CS290** **Workshop** **(1)**	**-**	**20**
III	**14CS310** **Probability and Statistics** **(3)**	**14CS320** **Theory and Design of Programming Languages** **(3)**	**14CS330** **Computer Graphics** **(3)**	**14CS341** **Engineering Design** **(3)**	**14CS350** **Data Structures and Algorithms** **(3)**	**--**	**14CS370** **Object Oriented Programming** **(3)**	**14CS380** **Data Structures Lab** **(1)**	**14CS390** **Assembly Language Programming Lab** **(1)**	**-**	**20**
IV	**14CS410** **Discrete Mathematics and Combinatorics** **(3)**	**14CS421** **System Software and Operating Systems** **(3)**	**14CS430** **Design and Analysis of Algorithms** **(3)**	**14CS440** **Database Management Systems** **(3)**	**14CS450** **Communication Engineering** **(3)**	**--**	**14CS470** **Professional Communication** **(2)**	**14CS480** **System Software and Operating Systems Lab** **(1)**	**15CS490** **Algorithms Lab** **(1)**	**14CS4C2** **Capstone Course – I** **(2)**	**21**
V	**14CS510** **Numerical Methods and Number Theory** **(3)**	**14CS521** **Computer Networks** **(3)**	**14CS530** **Theory of Computation** **(3)**	**14CS540** **Computer Architecture** **(3)**	**14CSPX0** ***Program Elective – I*** **(3)**	**--**	**14CS571** **Software Engineering : Theory and Practice** **(3)**	**14CS580** **Databases Lab** **(1)**	**14CS590** **Network Programming Lab** **(1)**	-	**20**
VI	**14CS610** **Project Management** **(3)**	**14CS620** **Internet Programming** **(3)**	**14CS630** **Compilers** **(3)**	**14CSPX0** **Program Elective – II** **(3)**	**14CSGX0** **General Elective - I** **(3)**	**--**	**14CS670** **Software Verification and Validation** **(3)**	**14CS680** **Internet Programming Lab** **(1)**	**14CS690** **Engineering by Design Lab** **(1)**	-	**20**
VII	**14CS710** **Accounting and Finance** **(3)**	**14CS720** **Artificial Intelligence** **(3)**	**14CS730** **Distributed Computing** **(3)**	**14CSOPX0** **Program Elective – III** **(3)**	**14CSPX0** **Program Elective – IV** **(3)**	**14CSGX0** **General Elective _ II** **(3)**	**--**	**14CS780** **Artificial Intelligence Lab** **(1)**	**--**	**14CS7C0** **Capstone Course – II** **(2)**	**21**
VIII	**14CSPXO** **Program Elective – V** **(3)**	**14CSPX0** **Program Elective – VI** **(3)**	**14CSPX0** **Program Elective VII** **(3)**	**--**	**--**	**--**	**--**	**14CS880** **Project** **(12)**		-	**21**

*For Reference only

B.E CSE - Programme Electives

Sl.No	Course Code	Course Name
1.	14CSPA0	Cryptography and Network Security
2.	14CSPB0	Data Warehousing and Mining
3.	14CSPC0	Wireless Networks
4.	14CSPD0	Enterprise Project Development using FOSS
5.	14CSPE0	Information Retrieval
6.	14CSPF0	Parallel Computing
7.	14CSPG0	Storage Infrastructure and Management
8.	14CSPH0	Service Oriented Architecture
9.	14CSPJ0	Design and Analysis of Algorithms - II
10.	14CSPK0	Real Time Operating Systems
11.	14CSPL0	User Interface Design
12.	14CSPM0	Cloud Computing and Virtualization
13.	14CSPN0	Internet of Things and its Applications
14.	14CSPP0	Big Data Analytics
15.	14CSPQ0	Soft Computing Techniques
16.	14CSPR0	Kernel Programming
17.	14CSPS0	Network Directory Services
18.	14CSPT0	Applied Machine Learning
19.	14CSPU0	Virtual Reality
20.	14CSPV0	Design Patterns

B.E CSE – One and Two Credit Courses

Sl.No	Course Code	Course Name
1.	14CS1A0	Introduction to Mainframe Systems
2.	14CS1B0	Mobile Application Development
3.	14CS1C0	Practical Approaches to Networking
4.	14CS1D0	Embedded Software Development
5.	14CS1E0	Introduction to IT Service Management
6.	14CS1F0	Green Data Center
7.	14CS1G0	Basics of Application Security
8.	14CS1H0	Foundations of NoSQL Database
9.	14CS1J0	Current Practices in Software Engineering
10.	14CS1K0	Agile Software Development & Safe
11.	14CS1L0	Heterogeneous Computing
12.	14CS2A0	Multicore Interconnects: Theory and Practice
13.	14CS2B0	Data Analytics using R and Python

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI- 625 015
(A Govt. Aided, ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
CHOICE BASED CREDIT SYSTEM

Degree: B.E., **Programme: Computer Science and Engineering**

1. **Compulsory Foundation Courses:** **Credits to be earned: (48 – 63)**

   a. **Humanities and Social Science (12 - 15)**

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14EG141	English	3	-	-	3	Nil
2.	14CS610	Project Management	3	-	-	3	Nil
3.	14CS710	Accounting and Finance	3	-	-	3	Nil
4.	14CS250	Environment Science	3	-	-	3	Nil
THEORY CUM PRACTICAL							
1.	14CS470	Professional Communication	1	-	2	2	14EG141
PRACTICAL							

   b. **Basic Science (15 - 21)**

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14MA110	Engineering Mathematics – I	2	2	-	3	Nil
2.	14CS210	Engineering Mathematics - II	2	2	-	3	Differentiation, integration and Elementary Calculus
3.	14CS310	Probability and Statistics	2	2	-	3	14MA110
4.	14CS410	Discrete Mathematics and Combinatorics	2	2	-	3	Basic Set Theory
5.	14PH120	Physics	3	-	-	3	Nil
6.	14CH130	Chemistry	3	-	-	3	Nil
THEORY CUM PRACTICAL							
PRACTICAL							
1.	14PH180	Physics Lab	-	-	2	1	Nil
2.	14CH190	Chemistry Lab	-	-	2	1	Nil

c. Engineering Science (15 - 21)

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14ES150	Basics of Civil and Mechanical Engineering	2	-	-	2	Nil
2.	14ES160	Basics of Electrical and Electronics Engineering	2	-	-	2	Nil
3.	14CS341	Engineering Design	3	-	-	3	Nil
4.	14CS220	Electronic Devices and Circuits	3	-	-	3	Nil
5.	14CS510	Numerical Methods and Number Theory	2	2	-	3	14MA110
THEORY CUM PRACTICAL							
1.	14ME170	Engineering Graphics	2	-	2	3	Nil
2.	14CS270	Problem Solving using Computers	2	-	2	3	Nil
PRACTICAL							
1.	14CS290	Workshop	-	-	2	1	14ES160

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI- 625 015
(A Govt. Aided, ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)

CHOICE BASED CREDIT SYSTEM

Degree: B.E **Programme: CSE**

d. Elective Foundation Courses: (HSS, BS or ES) **Credits to be earned: 06**

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14MAFA0	Graph Theory	3	-	-	3	Nil
2.	14MAFB0	Fuzzy Sets And Clustering	3	-	-	3	Nil
3.	14MAFC0	Number Theory	3	-	-	3	Nil
4.	14MAFD0	Mathematical Aptitude Techniques	3	-	-	3	Nil
5.	14PHFA0	Smart Materials	3	-	-	3	Nil
6.	14PHFB0	Thinfilm Technology	3	-	-	3	Nil
7.	14PHFC0	Nanotechnology	3	-	-	3	Nil
8.	14PHFD0	Quantum Computing	3	-	-	3	Nil
9.	14CHFA1	Biology For Engineers	3	-	-	3	Nil
10.	14CHFB0	Chemistry of Engineering Materials	3	-	-	3	Nil
11.	14CHFC0	Battery Technologies	3	-	-	3	Nil
12.	14CHFD0	Corrosion Science and Engineering	3	-	-	3	Nil
THEORY CUM PRACTICAL							
PRACTICAL							

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI- 625 015
(A Govt. Aided, ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)

CHOICE BASED CREDIT SYSTEM

Degree: B.E **Programme: CSE**

2. Core Courses: **Credits to be earned: (63 – 72)**

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14CS230	Digital Circuits	3	-	-	3	14CS160
2.	14CS241	Computer Organization and Microprocessors	3	-	-	3	14CS160
3.	14CS320	Theory and Design of Programming Languages	3	-	-	3	14CS270
4.	14CS330	Computer Graphics	3	-	-	3	14CS270
5.	14CS350	Data Structures and Algorithms	3	-	-	3	14CS270
6.	14CS421	System Software and Operating Systems	3	-	-	3	14CS240
7.	14CS430	Design and Analysis of Algorithms	3	-	-	3	14CS270, 14CS350
8.	14CS440	Database Management Systems	3	-	-	3	Nil
9.	14CS450	Communication Engineering	3	-	-	3	14CS230
10.	14CS4C2	Capstone Course - I	-	-	2	2	Nil
11.	14CS521	Computer Networks	3	-	-	3	Nil
12.	14CS530	Theory of Computation	3	-	-	3	14CS410
13.	14CS540	Computer Architecture	3	-	-	3	14CS230, 14CS240
14.	14CS620	Internet Programming	3	-	-	3	Nil
15.	14CS630	Compilers	3	-	-	3	14CS240, 14CS390, 14CS530
16.	14CS720	Artificial Intelligence	3	-	-	3	14CS410, 14CS310
17.	14CS730	Distributed Computing	3	-	-	3	14CS421, 14CS540,

							14CS521
18.	14CS7C0	Capstone Course - II	-	-	2	2	Nil
THEORY CUM PRACTICAL							
1.	14CS370	Object Oriented Programming	2	-	2	3	14CS270
2.	14CS571	Software Engineering: Theory and Practice	2	-	2	3	Nil
3.	14CS670	Software Verification and Validation	2	-	2	3	Nil
PRACTICAL							
1.	14CS280	Digital Circuits Lab	-	-	2	1	14ES160
2.	14CS380	Data Structures lab	-	-	2	1	14CS270
3.	14CS390	Assembly Language Programming Lab	-	-	2	1	14CS240
4.	14CS480	System Software and Operating Systems Lab	-	-	2	1	14CS240
5.	15CS490	Algorithms Lab	-	-	2	1	14CS270, 14CS350, 14CS380
6.	14CS580	Databases Lab	-	-	2	1	14CS440
7.	14CS590	Network Programming Lab	-	-	2	1	Nil
8.	14CS680	Internet Programming Lab	-	-	2	1	Nil
9.	14CS690	Engineering by Design Lab	-	-	2	1	Nil
10.	14CS780	Artificial Intelligence Lab	-	-	2	1	14CS720

3. Elective Courses: (27 - 39)

a. Programme Specific Elective **Credits to be earned:12 – 15**

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14CSPA0	Cryptography and Network Security	3	-	-	3	14CS521
2.	14CSPB0	Data warehousing and Mining	3	-	-	3	14CS440

3.	14CSPC0	Wireless Networks	3	-	-	3	14CS521
4.	14CSPD0	Enterprise Project Development using FOSS	3	-	-	3	14CS421
5.	14CSPF0	Parallel Computing	3	-	-	3	14CS430, 14CS540
6.	14CSPL0	User Interface Design	3	-	-	3	Nil
7.	14CSPM0	Cloud computing and virtualization	3	-	-	3	14CS370, 14CS521
8.	14CSPN0	Internet of Things and its Applications	3	-	-	3	14CS521, 14CS270
9.	14CSPQ0	Soft Computing Techniques	3	-	-	3	14CS270, 14CS430
10.	14CSPR0	Kernel Programming	3	-	-	3	14CS421
11.	14CSPS0	Network Directory Services	3	-	-	3	Nil
12.	14CSFB0	Data Communication	3	-	-	3	Nil
13.	14CSFC0	Object oriented Analysis and Design	3	-	-	3	Nil
THEORY CUM PRACTICAL							
PRACTICAL							

b. Programme Specific Elective for Expanded Scope

Credits to be earned: 09 – 12

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14CSPE0	Information Retrieval	3	-	-	3	14CS350
2.	14CSPG0	Storage Infrastructure Management	3	-	-	3	14CS521
3.	14CSPH0	Service oriented architecture	3	-	-	3	14CS521, 14CS620
4.	14CSPJ0	Design and Analysis of Algorithms – II	3	-	-	3	14CS350, 14CS430
5.	14CSPK0	Real Time Operating Systems	3	-	-	3	14CS421
6.	14CSPP0	Big Data Analytics	3	-	-	3	14CS440, 14CSPB0
7.	14CSPT0	Applied Machine Learning	3	-	-	3	14MA110 14CS210 14CS270 14CS310

							14CS410
8.	14CSPU0	Virtual Reality	3	-	-	3	14CS330
9.	14CSPV0	Design Patterns	3	-	-	3	Nil
One Credit Courses							
1.	14CS1A0	Introduction to Mainframe Systems	1	-	-	1	Nil
2.	14CS1B0	Mobile Application Development	1	-	-	1	14CS270, 14CS370
3.	14CS1C0	Practical Approaches to Networking	1	-	-	1	Basics of Networks
4.	14CS1D0	Embedded Software Development	1	-	-	1	14CS240
5.	14CS1E0	Introduction to IT Service Management	1	-	-	1	Nil
6.	14CS1F0	Green Data Center	1	-	-	1	Basic knowledge in Computer Architecture and Computing
7.	14CS1G0	Basics of Application Security	1	-	-	1	14CS620 14CS370
8.	14CS1H0	Foundations of NoSQL Database	1	-	-	1	14CS370 14CS440
9.	14CS1J0	Current Practices in Software Engineering	1	-	-	1	Knowledge of Software Engineering
10.	14CS1K0	Agile Software Development & Safe	1	-	-	1	14CS571 - Software Engineering: Theory and Practice
11.	14CS1L0	Heterogeneous Computing	1	-	-	1	C Programming
Two Credit Course							
1.	14CS2A0	Multicore Interconnects: Theory and Practice	2	-	-	2	14CS540
2.	14CS2B0	Data Analytics using R and Python	2	-	-	2	14CS310, Matrices, Vectors, Linear Algebra, Proficiency in SQL and basic structured data manipulation
PRACTICAL							

c. Interdisciplinary Elective **Credits to be earned: 06 – 12**

i. General Electives (Courses Offered by CSE Department)
Credits to be earned: 06

S.No	Course Code	Name of the Course	**Number of Hours / Week**			Credit	Prerequisite
			L	T	P		
THEORY							
1.	14CSGA0	Web Technologies	3	-	-	3	Nil
2.	14CSGB0	Essentials of Mobile Application Development	3	-	-	3	Basics of Computer Networks and Object oriented programming
3.	14CSGC0	Animation: Theory and Practice	3	-	-	3	Problem Solving using Computers
4.	14CSGD0	Essentials of Information Technology	3	-	-	3	Nil
5.	14CSGE0	Object Oriented Concepts and Design	3	-	-	3	Basic of Structured programming like C language
6.	14CSGF0	Enterprise Application Development	3	-	-	3	Nil
7.	14CSGG0	Programming using Python	3	-	-	3	Nil
THEORY CUM PRACTICAL							
PRACTICAL							

4. Project (12)

5. Skill/Proficiency based Elective **Credits to be earned: 02 - 04**

Total Credits – 164 (from 1 to 4) and 2 (from 5)

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

SECOND SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY
(For the candidates admitted from 2016 -17 onwards)

SECOND SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS210	Engineering Mathematics - II	BS	2	2	-	3
14CS220	Electronic Devices and Circuits	ES	3	-	-	3
14CS230	Digital Circuits	PC	3	-	-	3
14CS241	Computer Organization and Microprocessors	PC	3	-	-	3
14CS250	Environment Science	HSS	3	-	-	3
THEORY CUM PRACTICAL						
14CS270	Problem Solving using Computers	ES	2	-	2	3
PRACTICAL						
14CS280	Digital Circuits Lab	PC	-	-	2	1
14CS290	Workshop	ES	-	-	2	1
Total			**16**	**2**	**6**	**20**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture is equivalent to 1 credit

2 Hours Tutorial is equivalent to 1 credit

2 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

SECOND SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS210	Engineering Mathematics - II	3	50	50	100	25	50
2	14CS220	Electronic Devices and Circuits	3	50	50	100	25	50
3	14CS230	Digital Circuits	3	50	50	100	25	50
4	14CS241	Computer Organization and Microprocessors	3	50	50	100	25	50
5	14CS250	Environment Science	3	50	50	100	25	50
THEORY CUM PRACTICAL								
7	14CS270	Problem Solving using Computers	3	50	50	100	25	50
PRACTICAL								
8	14CS280	Digital Circuits Lab	3	50	50	100	25	50
9	14CS290	Workshop	-	100	-	100	-	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

		Category	L	T	P	Credit
14CS210	**ENGINEERING MATHEMATICS - II**	BS	2	1	0	3

Preamble

Vector calculus is a form of mathematics that is focused on the integration of vector fields. An Engineer should know the Transformations of the Integrals, as Transformation of Line Integral to surface and then to volume integrals. The Laplace transform method is a powerful method for solving linear ODEs and corresponding initial value problems as well as systems of ODEs arising in Engineering. The knowledge of transformations is to create a new domain in which it is easier to handle the problem that is being investigated. Complex Integration approach is very useful to evaluate many improper integrals of a real variable.

Prerequisite

Differentiation, Integration and Elementary calculus.

Course Outcomes

On the successful completion of the course, students will be able to

Demonstrate the use of double and triple integrals to compute area and volume (CO1).	Understand
Apply the concepts of Differentiation and Integration to Vectors.(CO2)	Apply
Apply Laplace transform technique to solve the given ordinary differential equation. (CO3) .	Apply
Compute an analytic function, when its real or Imaginary part is known. (CO4) .	Apply
Identify the Singularities and its corresponding Residues for the given function (CO5)	Understand
Predict a suitable method to evaluate the Contour integration (CO6)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	L											
CO2.	S	M										
CO3.	S	M										
CO4.	S	M										
CO5.	L											
CO6.	L											

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	40	20	60	40
Apply	40	60	20	40
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Predict the value of $\int_0^1\int_0^2\int_0^1 dxdydz$
2. Estimate the area enclosed by the curves $y = x^2$ and $\mathrm{x + y = 2}$
3. Predict the limits of $\int_0^a\int_0^x f(x, y)dydx,$ by changing the order of integration.

Course Outcome 2 (CO2):

1. The temperature of points in space is given by T = $x^2 + y^2$ - z , a mosquito located at (1,1,2) desires to fly in such a direction that it will get warm as soon as possible. In what direction shouls it move.
2. Verify Greens theorem for $\int_C \{[3x - 8y^2]dx + [4y - 6xy]dy\}$ where C is the boundary of the region bounded by x = 0, y=0 and x+y=1.
3. If A and B are irrotational , prove that A x B is solenoidal.

Course Outcome 3 (CO3):

1. Solve the Equation $y'' + 9y = \cos 2t,\ \mathrm{y(0)=1\ \&\ y}\left(\frac{\pi}{2}\right) = -1$ using Laplace Transform.
2. Compute $\mathrm{L}^{-1}\left(\frac{p+8}{p^2 + 4p + 5}\right)$
3. Using convolution theorem in Laplace Transform, evaluate $\int_0^t Sin\, u\ \ Cos(t-u)du$

Course Outcome 4 (CO4)

1. Compute an analytic function f(z)=u+iv, where $u = e^x(x\cos y - y\sin y)$.
2. Using convolution theorem, compute $L^{-1}\left(\frac{s^2}{(s^2 + a^2)(s^2 + b^2)}\right)$
3. Show that the map w = 1/z maps the circles and straight lines as circles or straight lines.

Course Outcome 5 (CO5) :

1. Define the term Residue of f(z) at z = a.
2. Identify the singular points of $\frac{1}{(2\sin z - 1)^2}$
3. Identify the residue of $\frac{z+1}{z^2 - 2z}$ at its poles

Course Outcome 6(CO6):

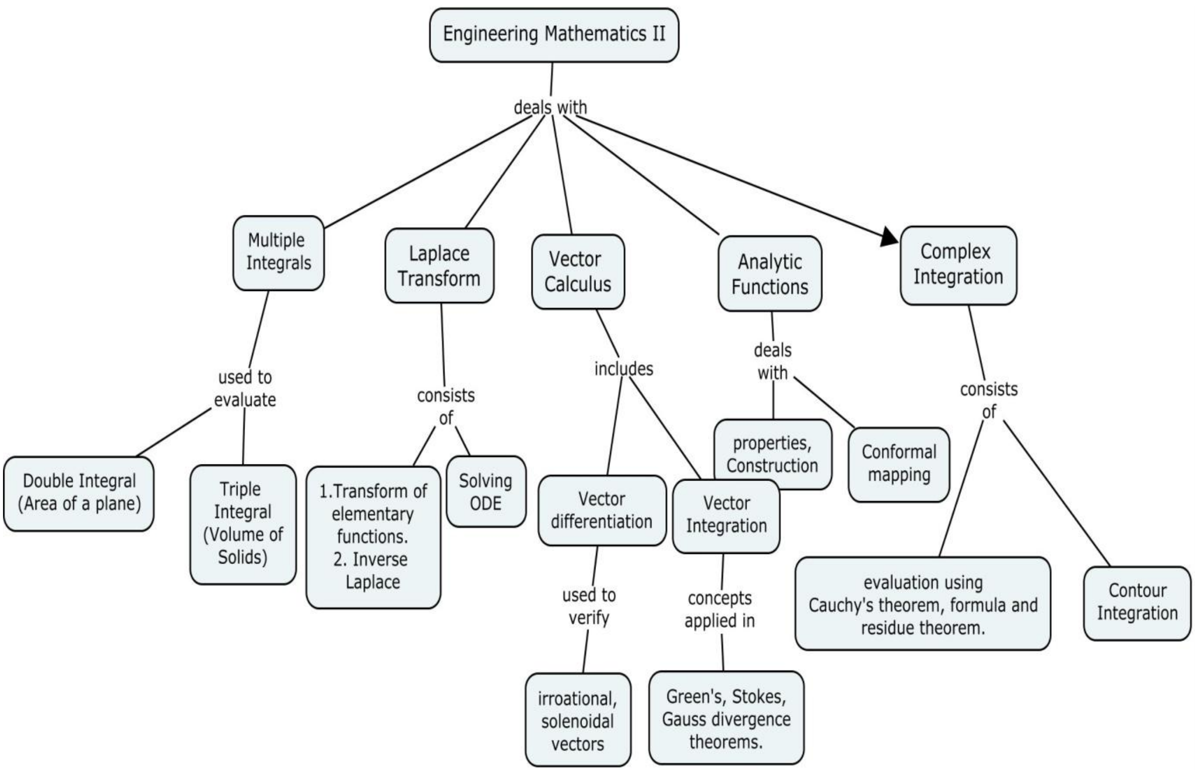
1. Evaluate $\int_{0}^{2\pi} \frac{Cos2\theta}{5+4Cos\theta} d\theta$

2. Examine the Laurent's series expansion of $f(z) = \frac{z+4}{(z+3)(z-1)^2}$,

   in (i) 0 < |z-1| < 4 (ii) |z-1| >

3. Evaluate $\int_{0}^{\infty} \frac{x \sin mx}{x^2 + a^2} dx$

Concept Map



Syllabus

MULTIPLE INTEGRALS: Double integrals –Change of order of integration –Double integrals in polar coordinates –Area enclosed by plane curves –Triple integrals –Volume of Solids –Change of variables in double and triple integrals**.**

VECTOR CALCULUS: Gradient, divergence and curl –Directional derivative –Irrotational and solenoidal vector fields –Simple problems on Vector differentiation–Vector integration –Green's theorem in a plane, Gauss divergence theorem and Stokes' theorem(excluding proofs)–Simple applications.

LAPLACE TRANSFORM : Laplace transform –Sufficient condition for existence –Transform of elementary functions –Basic properties –Transforms of derivatives and integrals of functions -Derivatives and integrals of transforms -Transforms of unit step function and impulse functions –Transform of periodic functions. Inverse Laplace transform -Statement of Convolution theorem –Initial and final value theorems–Solution of linear ODE of second order with constant coefficients using Laplace transformation techniques. **ANALYTIC FUNCTIONS :** Functions of a complex variable –Analytic functions: Necessary conditions –Cauchy -Riemann equations and sufficient conditions (excluding proofs) –Harmonic and

orthogonal properties of analytic function –Harmonic conjugate –Construction of analytic functions –Conformal mapping: $w = z^2$, sinz, e^z and bilinear transformation.
COMPLEX INTEGRATION : Complex integration –Statement and applications of Cauchy's integral theorem and Cauchy's integral formula –Taylor's and Laurent's series expansions – Singular points –Residues –Cauchy's residue theorem –Evaluation of real definite integrals as contour integrals around unit circle and semi-circle (excluding poles on the real axis).

Text Books

1. Grewal. B.S, "Higher Engineering Mathematics", 41st Edition, Khanna Publications, Delhi, 2011.
2. Erwin Kreyszig, Advanced Engineering Mathematics, 8th Edition, John Wiley & Sons, 2009.

Reference Books

1. T.Veerarajan, Engineering Mathematics, 3rd Edition, Tata McGraw Hill, New Delhi, 2004.
2. Thomas Phinny, Calculus, 13th Edition, Pearson Education, New Delhi,2005.
3. B.V.Ramana, Higher Engineering Mathematics, Tata Macraw Hill, New Delhi,2011

Course Contents and Lecture Schedule

Module No.	Topic	No.of Lectures
1	**Multiple Integrals**	
1.1	Double integrals and areas	1
1.2	Triple integrals and volumes	1
	Tutorial	2
1.3	Change of order of integration	1
1.4	Change of variables between Cartesian and polar with applications	2
	Tutorial	2
2	**Vector Calculus**	
2.1	Operators Grad, div and curl with properties	1
	Tutorial	1
2.2	Solenoidal and irrotational vectors	2
	Tutorial	2
2.3	Vector integration(Green,Gauss,Stokes)	2
	Tutorial	2
3	**Laplace Transformation**	
3.1	Laplace transformation-properties, inverse laplace transforms	2
	Tutorial	2
3.2	Periodic functions, convolution theorem, initial value theorem and final value theorem	1
	Tutorial	2
3.3	Solution of differential equations and integral equations	1
	Tutorial	2
4	**Analytic Functions**	
4.1	Analytic functions, C-R equations and properties	1
	Tutorial	1
4.2	Harmonic functions and Milne Thomson's method	2
	Tutorial	2

4.3	Conformal maps and bilinear transformations	2
	Tutorial	2
5	**Complex Integration**	
5.1	Cauchy's theorem	1
5.2	Evaluating integrals using Cauchy's integral formula	1
	Tutorial	2
5.3	Taylor's and Laurent's expansions	1
5.4	Singularities and Cauchy's residue theorem	1
	Tutorial	2
5.5	Contour integration using unit circle and semicircular contours	1
	Total	**48**

Course Designers:

1. Dr.S.Jeyabharathi sjbmat@tce.edu
2. Dr. G. Jothilakshmi gjlmat@tce.edu
3. Dr. A.P.Pushpalatha appmat@tce.edu

		Category	L	T	P	Credit
14CS220	**ELECTRONIC DEVICES AND CIRCUITS**	ES	3	0	0	3

Preamble

The course work aims in imparting fundamental knowledge of semiconductors required for computer science engineers. The course work will introduce engineers to different types of semiconductor devices. The structure, operation, characteristics and applications of various type of transistors are discussed. The fundamental ideas of logic gates and Boolean laws are given in the course work.

Prerequisite

Basic course (No prerequisite)

Course Outcomes

On the successful completion of the course, students will be able to

Illustrate Fermi level, mobility, carrier concentration, band gap P&N type semiconductors. (CO1)	Understand
Compute the conductivity, mobility, carrier concentration of Intrinsic and extrinsic semiconductors. (CO2)	Apply
Explain the characteristics and application of diodes for rectification (CO3)	Understand
Interpret the modes of configuration of transistors and their performance. (CO4)	Apply
Distinguish the working principles and applications of different types of electronic devices. (CO5)	Understand
Explain the functions and applications of Logic gates.(CO6)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M											
CO2	S											
CO3	M											
CO4	S											
CO5	M											
CO6	M											

S- Strong; M-Medium; L-Low ;

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	50	50	50	50
Apply	30	30	30	30
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define Fermi level
2. Estimate the drift velocity of electron in an electric field of 300 V/m, if the electron mobility is 0.52 m^2/V -s
3. List any two factors affecting the band gap of a semiconductor.

Course Outcome 2 (CO2):

1. Compare Intrinsic & Extrinsic Semiconductors (Remember).
2. Calculate the carrier concentration in n type semiconductors. (Understand).
3. Determine the values of I_C and I_E for a transistor with α_{dc} = 0.97 and I_{CBO} =1o µA and I_β = 50 mA (Apply)

Course Outcome 3 (CO3):

1. Comment on the V – I characteristics of Diode.(Understand).
2. A FET has a drain current of 4 mA. If I_{DSS} = 8mA and $I_{GS\ off}$ = -6V. Find the values of V_{GS} and V_p (Apply)
3. Compare half wave rectifier with Bridge rectifier.(Understand)

Course Outcome 4 (CO4):

1. The leakage current of the transistor with usual notations are I_{CEO} = 410 mA , I_{CEO} = 410 mA and I_{CBO} = 30 mA . Calculate I_C (Understand)
2. Construct the family of common source static drain characteristics of n channel JFET and discuss (Understand)
3. Consider the self bias circuit where V_C =22.5 v, R_C = 5.6 kΩ, R_e = 1 kΩ, R_2 = 10 kΩ and R_1 = 90kΩ, h_{fe} = 55 = β; V_{be} = 0.6 V.The transistor operates in active region. Determine the Q point of the transistor and its stability factor.

Course Outcome 5 (CO5):

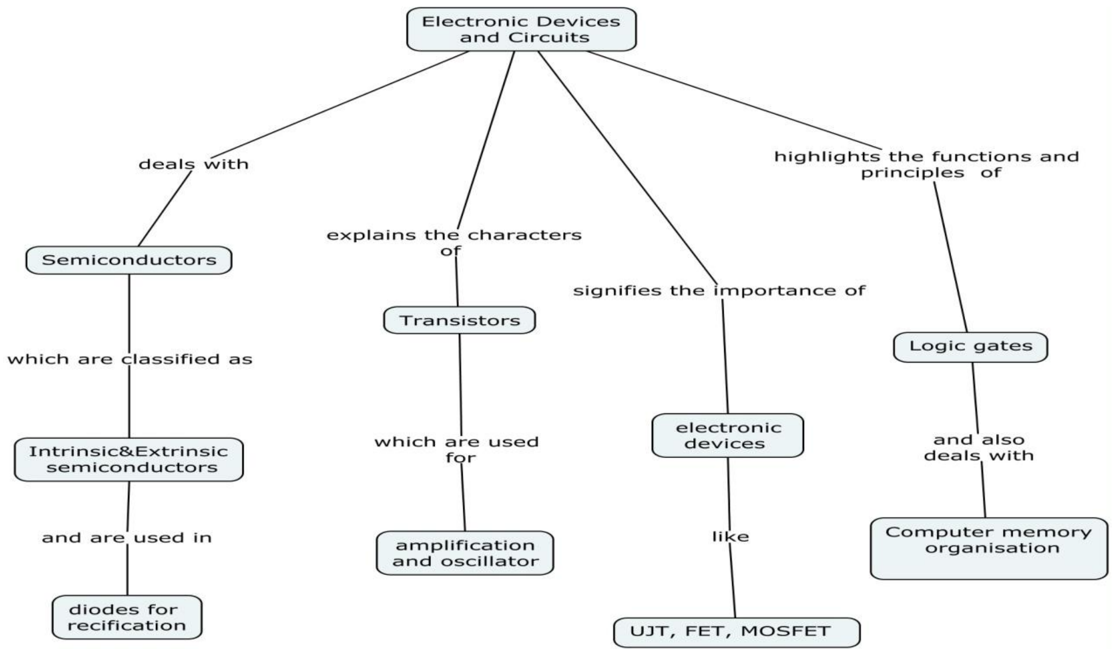
1. Demonstrate the working principle of SCR and explain its VI characteristics.(Understand)
2. Determine V_{GSQ} , I_{DQ} , V_D, v_G, v_s, V_{DS} for the common gate configuration of FET by assuming the following values for R_D=1.5 kΩ, R_S =680 Ω, I_{DSS} =12 mA, V_P=-6 v.(Understand)
3. Construct a Wien Bridge oscillator. (Understand)

Course Outcome 6 (CO6):

1. Draw the logic diagram to implement the following Boolean expression using NAND and NOR gate. $X = \overline{A}(B+C) + D$
2. Differentiate between EXCLUSIVE OR and INCLUSIVE OR gate.
3. Classify the following as positive or negative logic

	0	1
a)	0 V	10V
b)	0.2 v	- 3.6V

Concept Map



Syllabus

Semiconductors - Intrinsic & Extrinsic Semiconductors – N & P type - Fermi level - mobility – conductivity – carrier concentration in n type semiconductors – Continuity equation – Hall effect – applications – Theory of P – N junction – junction diode – V – I characteristics of Diode – Diode as a switch – Transistor and diffusion capacitance of Diode – Breakdown of diode – Rectifier (HW, FW & Bridge)
Electronic devices –Transistors- Structure, Operation – three modes of configuration –– Currents in Transistor – Relation between α & β – load line – Transistor as an amplifier (E) – A_v, A_p & η of transistor – Transistor as an oscillator – Feedback mechanism – condition for oscillations – switching action of transistor- UJT, FET, MOSFET - Structure, Operation and characteristics – - Applications - Fabrication of ICs
Digital circuits and memory systems-Binary concept –Logic gates -De Morgan's laws- NAND & NOR logic gates-DTL NAND & RTL NOR gates-Implementation of logic circuits with NAND/NOR gates-Implementation of logic circuits for Boolean expressions- Organization of Computer memories- Memory system design-hierarchy & characteristics- Classification of memories.

Text Books

1. David A Bell, Electronic Devices And Circuits, PHI, 2007
2. Malvino, Electronic Devices And Circuits, PHI, 2007
3. Allen Mottershed, Electronic Devices And Circuits, PHI,2009

Reference Books

1. Robert L.Boylestad, Louis Nashelsky, Electronic Devices And Circuit Theory, PHI

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Semiconductors**	
1.1	Intrinsic & Extrinsic Semiconductors	1

Module No.	Topic	No. of Lectures
1.2	N & P type semiconductor	1
1.3	Fermi level - mobility – conductivity	1
1.4	carrier concentration in n type semiconductors	1
1.5	Continuity equation	2
1.6	Hall effect - applications	1
1.7	Theory of P – N junction – junction diode – V – I characteristics of Diode	1
1.8	Diode as a switch – Transistor and diffusion capacitance of Diode – Breakdown of diode	2
1.9	Rectifier (HW, FW & Bridge)	2
2.	**Electronic devices**	
2.1	Transistor-Structure & Operation of three modes of configuration	2
2.2	Currents in Transistor – Relation between α & β	1
2.3	load line analysis	1
2.4	Transistor as an amplifier (E) – A_v, A_p & η of transistor	1
2.5	Transistor as an oscillator	1
2.6	Feedback mechanism – condition for oscillations – switching action of transistor	2
2.7	UJT, FET, MOSFET- Structure, Operation and characteristics	3
2.8	Fabrication of ICs	1
3.	**Digital circuits and memory systems**	
3.1	Binary concept –Logic gates -De Morgan's laws-Implementation of logic circuits for Boolean expressions-	3
3.2	NAND & NOR logic gates	2
3.3	DTL NAND & RTL NOR gates-Implementation of logic circuits with NAND/NOR gates	2
3.4	Organization of Computer memories	1
3.5	Memory system design- hierarchy & characteristics	2
3.6	Classification of memories	2
	Total	**36**

Course Designers:

1. Dr. R. Vasuki rvphy@tce.edu

		Category	L	T	P	Credit
14CS230	**DIGITAL CIRCUITS**	PC	3	0	0	3

Preamble

The syllabus is designed for the students to learn and understand the basic principles of number systems, binary arithmetic , Boolean algebra and digital logic gates and circuits. It illustrates different methods for simplification of Boolean logic functions. These methods include algebraic simplification, karnaugh maps and Quine McClusky tabulation technique. Then the principles of combinational logic circuits, their design and implementation are demonstrated. The fundamental concepts of synchronous sequential logic circuits, starting from different flip flops and their design techniques are exemplified. A brief introduction to Hardware Design Language and its use in simulation of combinational and sequential logic circuits are also provided. Next, an introduction to the configuration of memory and programmable logic and their application in implementing combinational logic circuits are presented.

Prerequisite

14ES160: Basics of Electrical and Electronics Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Apply the principles of number systems and binary codes to carry out arithmetic and code conversions. (CO1)	Apply
Apply the theorems and postulates of Boolean algebra, the techniques of Karnaugh Maps and Quine McClusky tabulation techniques for simplification of logic functions.(CO2)	Apply
Design combinational logic circuits for various applications and implement them using logic gates or other devices like multiplexers, decoders or programmable logic devices and simulate them using Hardware Description Language (HDL). (CO3)	Apply
Design sequential logic circuits like counters and sequence detectors and implement them using different flip flops. (CO4)	Apply
Analyse the given combinational or sequential logic circuit to determine its function. (CO5)	Analyse

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L									
CO2	S	M	L									
CO3	S	M	M	L	M							
CO4	S	M	M	L								
CO5	S	S		L								

S- Strong; M-Medium; L-Low ;

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	Theory
Remember	20	10	10	10
Understand	40	20	20	20
Apply	40	50	50	50
Analyse	-	20	20	20
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Apply the principles of 2's complement notation to solve 87-59, 20-94, -56-78, 99+87 (Apply)
2. State the need for BCD .(Remember)
3. Explain the drawback of 1's complement form for representing signed binary numbers.(Understand)
4. Solve the conversion of the binary fraction 110011.011 to decimal (Apply)
5. Solve the conversion of the decimal number 65979 to binary using hexadecimal as intermediate form. (Apply)

Course Outcome 2 (CO2):

1. Utilise the theorems of Boolean algebra to simplify f = a'bc'+a'b'+abc' (Apply)
2. Define a prime implicant.(Remember)
3. Make use of K map to obtain minimal POS form of g = $\sum m(0,2,4,6,9,12,14)$ (Apply)
4. Construct the simplified SOP form of h = $\sum m(3,5,7,9,13,15,17) + \sum d(12,14)$ using QM tabulation. (Apply)
5. Illustrate the drawbacks of algebraic and Karnaugh map simplification (Understand)

Course Outcome 3 (CO3):

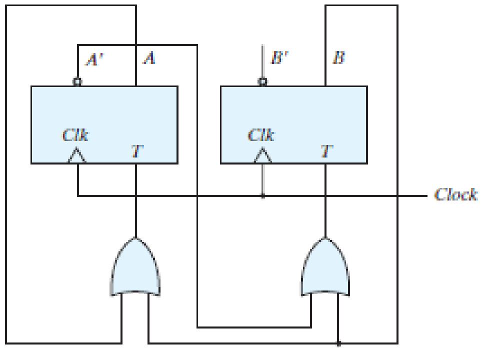
1. Design a 2- bit magnitude comparator to compare two binary numbers. (Apply)
2. Design a 4-input priority encoder (Apply)
3. Develop the verilog description of a full adder. (Apply)
4. State the functions of a demultiplexer. (Remember)
5. Differentiate between PLA and PAL.(Understand)
6. Develop the verilog description of a 4 to 1 multiplexer. (Apply)
7. Construct the ROM implementation of G = $\sum m(0,1,2,4,7,9,12)$ (Apply)
8. Make use of a 4 to 1 multiplexer to implement the function h = $\sum m(2,4,6,7)$ (Apply)
9. Make use of a 4 to 16 decoder to implement f = $\sum m(3,5,7,8,9,15)$ (Apply)
10. Construct the PLA implementation of the functions f = AB'+AC+A'BC' and
    a. G = (AC+BC)' (Apply)

Course Outcome 4 (CO4):

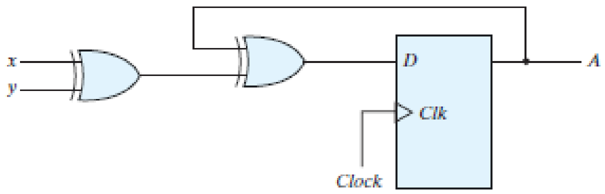
1. Compare a latch and a flip flop (Understand)
2. Design a 2-bit synchronous up counter using JK flip flops. (Apply)
3. Make use of a D flip flop to construct a JK flip flop (Apply)
4. Design a 4-bit ripple down counter .(Apply)
5. Design a Moore type sequence detector to detect the input sequence 101. (Apply)
6. List the drawback of SR flip flop. (Remember)
7. Explain the need for edge triggering.(Understand).

Course Outcome 5 (CO5):

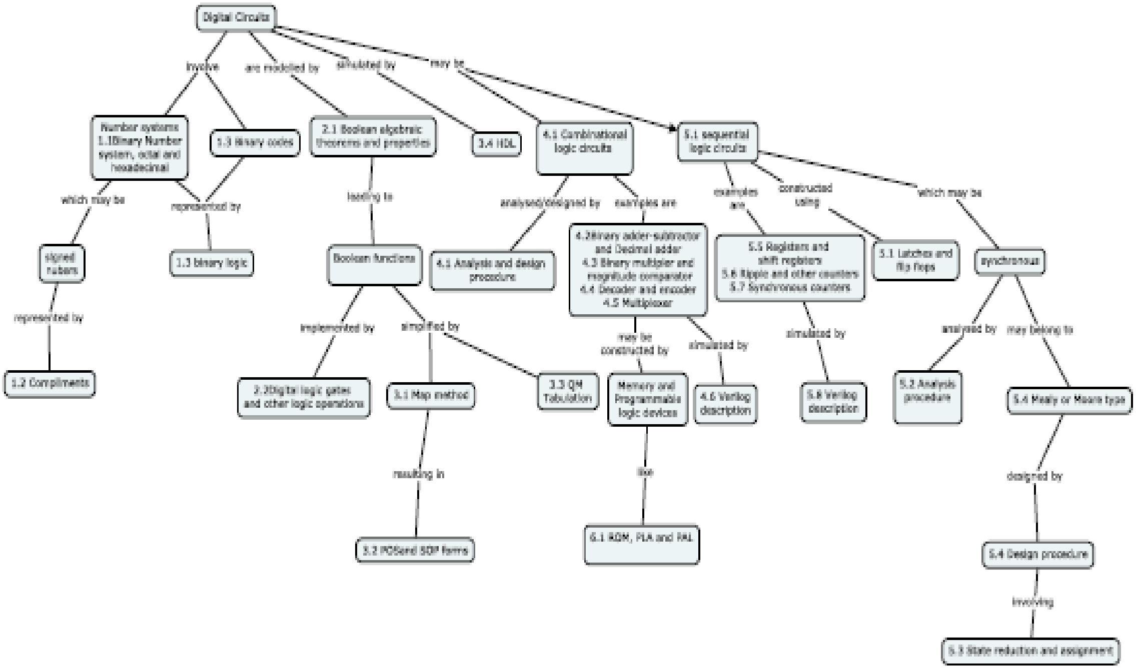
1. Analyse the following circuit, draw the state diagram and examine the function of the circuit. (Analyse)



2. Analyse the following circuit, draw the state diagram and examine its operation. (Analyse)



Concept Map



Syllabus

Number Systems:

Binary Numbers, Octal and Hexadecimal Numbers and conversions, Complements - Signed Binary Numbers, Binary Codes and Binary Logic.

Boolean Algebra and Logic Gates :

Basic Definition , Theorems and Properties of Boolean Algebra, Boolean functions, Digital Logic Gates and Other Logic Operations.

Gate Level Minimization:

The Karnaugh Map Method – Three and Four Variable Maps, Product of Sums(POS) simplification including don't care conditions, Quine-McCluskey (QM) Technique, Introduction to Hardware Description Language.

Combinational Logic :

Introduction to Combinational Circuits, Analysis and Design Procedure, Binary Adder-Subtractor and Decimal Adder, Binary multiplier, Binary magnitude comparator, Decoders Encoders and Priority Encoders, Multiplexers and their applications, Verilog description of combinational logic circuits.

Sequential Logic Circuits :

Introduction to Sequential Circuits, Latches and flip-flops, Analysis Procedure for clocked sequential circuits, State reduction and assignment, Mealy and Moore machines and their design procedure, Registers and shift registers, Ripple and other Counters, Synchronous Counters, Verilog description of sequential logic circuits.

Memory and Programmable Logic :

Organization of ROM, PLA and PAL and their application in implementing combinational logic circuits.

Text Books

1. M.Morris Mano & Michael D.Ciletti, Digital Design, First impression,Pearson, 2012.

Reference Books

1. Mohammed Ferdjallah, Introduction to digital systems, Modelling, Synthesis and simulation using VHDL, Wiley,2011.
2. D.P. Leach & A.P. Malvino, Digital Principles and Applications, sixth edition, Tata McGraw Hill,2006.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Number Systems:**	
1.1	Binary Numbers, Octal and Hexadecimal Numbers and conversions	1
1.2	Complements - Signed Binary Numbers	1
1.3	Binary Codes and binary logic	1
2	**Boolean Algebra and Logic Gates :**	
2.1	Basic Definition , Theorems and Properties of Boolean Algebra and Boolean functions	2
2.2	Digital Logic Gates and Other Logic Operations	1
3	**Gate Level Minimization:**	
3.1	The Karnaugh Map Method – Three and four Variable Maps	2
3.2	Product of Sums(POS) simplification including don't care conditions.	1
3.3	Quine-McCluskey(QM) Technique	2
3.4	Introduction to Hardware Description Language	2
4	**Combinational Logic :**	
4.1	Introduction to Combinational Circuits , analysis and design procedure	1
4.2	Binary Adder-Subtractor and decimal adder	1
4.3	Binary multiplier and magnitude comparator	1
4.4	Decoders, Encoders and Priority encoders	1
4.5	Multiplexers and their applications	1
4.6	Verilog description of combinational logic circuits	2
5	**Sequential Logic Circuits :**	
5.1	Introduction to sequential circuits, Latches and flip-flops	2
5.2	Analysis Procedure for clocked sequential circuits	1
5.3	State reduction and assignment.	1
5.4	Mealy and Moore machines and their Design Procedure.	3
5.5	Registers and shift registers	1
5.6	Ripple and other Counters	1
5.7	Synchronous Counters	1
5.8	Verilog description of sequential logic circuits	3
6	**Memory and Programmable Logic :**	
6.1	Organization of ROM, PLA and PAL and their application in implementing combinational logic circuits.	3

Course Designers:

1. Mr. C.Sridharan cscse@tce.edu
2. Dr. N.Balamurugan nbbalamurugan@tce.edu

14CS241	COMPUTER ORGANIZATION AND MICROPROCESSORS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

The syllabus is designed for the students to learn and understand the basic organization of computers and the working of its functional components. It gives a brief overview of the organisation of a computer, simple Von Neumann machine organisation of IAS computer, memory format and instruction execution in it. Then memory hierarchy, types of memories, organisation of main memory, types of IO buses, and their operation and timing diagrams are presented. Then the elements of cache memory design, its mapping functions and replacement algorithms are emphasised followed by performance estimation of disk drives under interrupt driven and DMA driven approaches are discussed. Then the architecture of Intel x86, its addressing modes, instruction set and elements of assembly language programming are illustrated, leading to development of simple assembly language programs.

Prerequisite

14ES160: Basics of Electrical and Electronics Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Explain the evolutions of computers, the organisational features, structure, memory format and operation of IAS computers, components of a computer and its instruction cycle with and without interrupts. (CO1) Understand

Estimate the data transfer rate and the length of the instruction cycle for synchronous read and write cycles, with an understanding of the bus interconnection and timing diagrams for synchronous and asynchronous buses. (CO2) Apply

Explain the features of main memory organisation and its types. (CO3) Understand

Design the cache memory organisation, its mapping functions and replacement algorithms and estimate the performance improvement. (CO4) Apply

Estimate the performance of disk devices and the consumption of CPU time due to interrupt driven and DMA data transfer. (CO5) Apply

Perform integer and floating point arithmetic operations on binary numbers. (CO6) Apply

Develop simple assembly language programs for arithmetic, code conversion and sorting operations, with an understanding of architecture and interrupt processing of Intel x86 CPUs. (CO7) Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	L	L										
CO2.	S	S	S	M								
CO3	M	S	M	M								
CO4	S	S	S	M	M							
CO5	M	S	M	M								
CO6	S	S	M	M								
CO7	S	S	S	S	S							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	40	40	30	30
Apply	30	40	50	50
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1:

1. List the basic functions a computer can perform. (Remember)
2. Distinguish between computer structure and computer function? (Understand)
3. List the features of a Von Neumann computer? (Remember)
4. Explain the two approaches to dealing with multiple interrupts. (Understand)
5. Explain the basic instruction cycle with a block diagram. (Explain)
6. Explain the instruction cycle with interrupts.(Understand)

Course Outcome 2:

1. Consider two microprocessors having 8- and 16-bit-wide external data buses, respectively. The two processors are identical otherwise and their bus cycles take just as long. Suppose all instructions and operands are two bytes long. By what factor do the maximum data transfer rates differ? Repeat assuming that half of the operands and instructions are one byte long. (Understand)
2. For a synchronous read operation, the memory module must place the data on the bus sufficiently ahead of the falling edge of the Read signal to allow for signal settling. Assume a microprocessor bus is clocked at 10 MHz and that the Read signal begins to fall in the middle of the second half of T3. Determine the length of the memory read instruction cycle. When, at the latest, should memory data be placed on the bus? Allow 20 ns for the settling of data lines. (Apply)
3. Consider a 32-bit microprocessor whose bus cycle is the same duration as that of a 16-bit microprocessor. Assume that, on average, 20% of the operands and instructions are 32 bits long, 40% are 16 bits long, and 40% are only 8 bits long. Calculate the improvement achieved when fetching instructions and operands with the 32-bit microprocessor. (Apply)
4. State the benefit of using a multiple-bus architecture compared to a single-bus architecture? (Remember)
5. Explain the timing diagram of asynchronous bus read and write cycles.(Understand)

Course Outcome 3:

1. List the properties of semiconductor memory cells. (Remember)
2. Explain the operation of a DRAM cell.(Understand)
3. List the applications ROM. (Remember)

4. How does SRAM differ from DRAM ? (Understand)
5. Explain the organisation of a typical 16-Mbit DRAM.(Understand)

Course Outcome 4:

1. For a direct-mapped cache, a main memory address is viewed as consisting of three fields. List and define the three fields. (Remember)
2. A set-associative cache consists of 64 lines, or slots, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Show the format of main memory addresses. (Understand)
3. Given the following specifications for an external cache memory: four-way set associative; line size of two 16-bit words; able to accommodate a total of 4K 32-bit words from main memory; used with a 16-bit processor that issues 24-bit addresses. Design the cache structure with all pertinent information and show how it interprets the processor's addresses. (Apply)
4. Describe a simple technique for implementing an LRU replacement algorithm in a four-way set-associative cache. (Understand)
5. Consider a cache of 4 lines of 16 bytes each. Main memory is divided into blocks of 16 bytes each. That is, block 0 has bytes with addresses 0 through 15, and so on. Now consider a program that accesses memory in the following sequence of addresses:
6. Once: 63 through 70 Loop ten times: 15 through 32; 80 through 95.a. Suppose the cache is organized as direct mapped. Memory blocks 0, 4, and so on are assigned to line 1; blocks 1, 5, and so on to line 2; and so on. Compute the hit ratio. b. Suppose the cache is organized as two-way set associative, with two sets of two lines each. Even-numbered blocks are assigned to set 0 and odd-numbered blocks are assigned to set 1. Compute the hit ratio for the two-way set-associative cache using the least recently used replacement scheme. (Apply)

Course Outcome 5:

1. Explain the components of a disk drive. (Understand)
2. Define rotational latency and access time. (Remember)
3. Consider a magnetic disk drive with 8 surfaces, 512 tracks per surface, and 64 sectors per track. Sector size is 1 kB. The average seek time is 8 ms, the track-to-track access time is 1.5 ms, and the drive rotates at 3600 rpm. Successive tracks in a cylinder can be read without head movement. a. What is the disk capacity?

   b. What is the average access time? Assume this file is stored in successive sectors and tracks of successive cylinders, starting at sector 0, track 0, of cylinder i.

   c. Estimate the time required to transfer a 5-MB file.

   d. What is the burst transfer rate? (Apply)
4. A particular system is controlled by an operator through commands entered from a keyboard. The average number of commands entered in an 8-hour interval is 60. Suppose the processor scans the keyboard every 100 ms. How many times will the keyboard be checked in an 8-hour period? By what fraction would the number of processor visits to the keyboard be reduced if interrupt-driven I/O were used? (Apply)
5. A DMA module is transferring characters to memory using cycle stealing, from a device transmitting at 9600 bps. The processor is fetching instructions at the rate of 1 million instructions per second (1 MIPS). By how much will the processor be slowed down due to the DMA activity? (Apply)

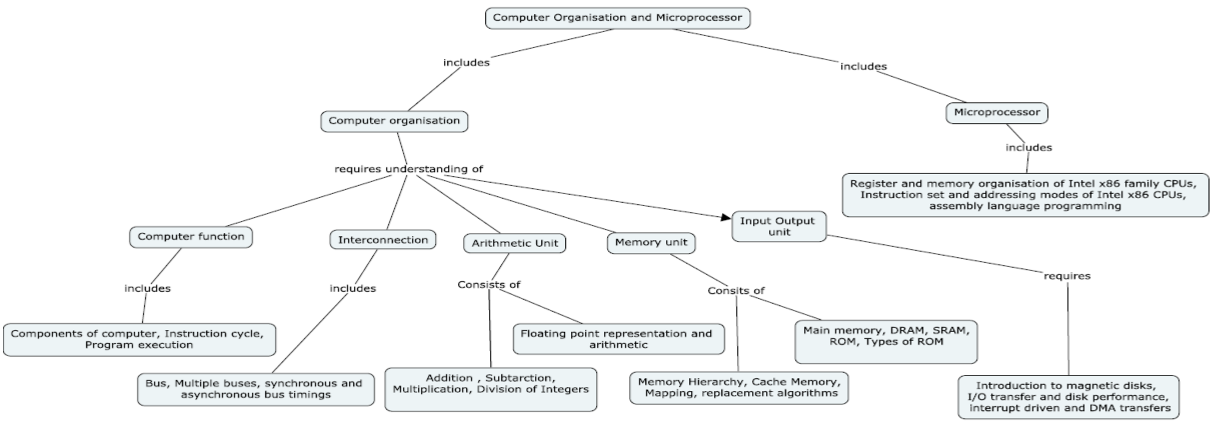
Course Outcome 6:

1. Explain the flow chart for unsigned binary multiplication. (Understand)
2. Explain a typical 32-bit floating point representation. (Understand)
3. State the range of n-bit 2's complement representation. (Remember)
4. Use the Booth algorithm to multiply 23 (multiplicand) by 29 (multiplier), where each number is represented using 6 bits. (Apply)
5. Illustrate how the following floating-point additions are performed (where significands are truncated to 4 decimal digits). Show the results in normalized form.
   $3.344 * 10^{1} + 8.877 * 10^{-2}$ (Apply)

Course Outcome 7:

1. State the purpose of pointer group of registers.(Remember)
2. State the purpose of addressing mode.(Remember)
3. Differentiate between carry and overflow flags. (Understand)
4. How is the effective address calculated in indexed addressing mode? (Understand)
5. Develop an assembly language program to perform addition of 32-bit numbers. (Apply)
6. Develop an assembly language program to perform addition of an array of 16-bit numbers.(Apply)
7. Develop an assembly language program to perform multiplication of 32-bit by 16-bit (Apply)
8. Develop an assembly language program for BCD to Excess 3 code conversion. (Apply)
9. Develop an assembly language program for sorting an array of 8-bit numbers.(Apply)

Concept Map



Syllabus

Introduction: Computer organisation and architecture, Von neumann machine, Evolution of computers and generations, Introduction to IAS computer structure and operation.

Computer Function and Interconnection: Top level view of components and functions, Instruction cycle and program execution, Interrupts and instruction cycles, multiple interrupts, Interconnection structures, Bus interconnection, multiple buses, Synchronous and asynchronous bus timings.
Memory and I/O : Characteristics and hierarchy of memory, Cache memory principles and operation, Cache design and mapping functions, replacement algorithms, main memory, DRAM and SRAM, Types of ROMs, Module organisation, Introduction to magnetic disks, I/O transfer and disk performance, interrupt driven and DMA transfers.
Computer Arithmetic : Arithmetic and Logic Unit, Integer multiplication of unsigned and signed numbers, Booth's algorithm, division of unsigned binary, floating point representation and arithmetic.
Microprocessors: Register and memory organisation of Intel x86 family CPUs, interrupt processing, elements of machine instructions, Instruction set and addressing modes of Intel x86 CPUs, assembly language programming.

Text Books

1. William Stallings, Computer Organization and Architecture Designing for Performance, Ninth edition, Prentice Hall, 2012.

Reference Books

1. Andrew S Tanenbaum and Todd Austin, Structured Computer Organization, Sixth edition, Pearson, 2013.
2. Carl Hamacher, Computer Organization and Embedded Systems, Sixth edition, McGrawHill, 2012.
3. Dodiya Tripti, Computer Organisation and Advanced Microprocessors, First edition, Cengage Learning India,2012.
4. Barry B.Brey, The Intel Microprocessors Architecture Programming and Interfacing, Eighth edition, Pearson Prentice Hall, 2009.
5. N.Senthil Kumar, M.Saravanan and S. Jeevananthan, Microprocessors and Microcontrollers, First edition, Oxford University Press, 2010.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction:**	
1.1	Computer organisation and architecture.	1
1.2	Evolution of generation of computers and Von Neuman machine.	1
1.3	Introduction to IAS computer structure and operation	2
2	**Computer Function and Interconnection:**	
2.1	Top level view of components and functions, Instruction cycle and program execution	1
2.2	Interrupts and instruction cycles, multiple interrupts	2
2.3	Interconnection structures, Bus interconnection, multiple buses	2
2.4	Synchronous and asynchronous bus timings.	2
3	**Memory and I/O:**	
3.1	Characteristics and hierarchy of memory, Cache memory principles and operation.	1
3.2	Cache design, mapping functions and replacement algorithms	3
3.3	Main memory, DRAM and SRAM	2

Module No.	Topic	No. of Lectures
3.4	Types of ROMs, Module organisation	1
3.5	Introduction to magnetic disks, I/O transfer and disk performance,	2
3.6	Interrupt driven and DMA transfers	2
4	**Computer Arithmetic:**	
4.1	Arithmetic and Logic Unit, Integer multiplication of unsigned and signed numbers, Booth's algorithm	3
4.2	Division of unsigned binary numbers	1
4.3	Floating point representation and arithmetic.	2
5	**Microprocessors :**	
5.1	Register and memory organisation of Intel x86 family CPUs	2
5.2	Interrupt processing	1
5.3	Elements of machine instructions, Instruction set and addressing modes of Intel x86 CPUs,	2
5.4	Assembly language programming.	3
	Total	**36**

Course Designers:

1. C.Sridharan (cscse@tce.edu)
2. Dr.P.Chitra (pccse@tce.edu)

14CS250	ENVIRONMENT SCIENCE	Category	L	T	P	Credit
		HSS	3	0	0	3

Preamble

This course provides the basic knowledge of structure and function of ecosystem and better understanding of natural resources, biodiversity and their conservation practices. It describes the need to lead more sustainable lifestyles, to use resources more equitably. It helps to create a concern for our environment that will trigger pro-environmental action, including activities we can do in our daily life to protect it. Furthermore, it deals the social issues and ethics to develop quality engineer in our country.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Explain the significance of the conservation of natural resources and Identify the significance of Environmental studies in the context of multiple disciplines. (CO1)	Understand
Demonstrate an understanding of different ecosystems and identify the influence of various factors in guiding the evolution of an ecosystem. (CO2)	Understand
Make use of an understanding of the types, values, hotspots of Bio-diversity and threats to Biodiversity in solving conflicts between organisms. (CO3)	Apply
Identify various causes, effects of environmental pollution and make use of various control measures to counteract the effects of pollution. (CO4)	Apply
Apply the environmental conservation concepts to achieve environment sustainability. (CO5)	Apply
Identify areas of Information and Communication Technology (ICT) that directly contribute to the Green House Gas Emissions and environmental pollution and develop solutions to reduce the adverse impact of ICT on the environment. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.						L	M					
CO2.						L	M					
CO3			L			M	S					
CO4			L			M	S					
CO5.			L			M	S					L
CO6		M	L			M	S					L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	40	30	30	30

Apply	40	50	50	50
Analyse				
Evaluate				
Create				

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the multidisciplinary nature of Madurai district
2. Explain the necessity of food web
3. Account for energy of pyramid of ecosystem always upright

Course Outcome 2 (CO2):

1. List out types of ecological succession
2. Demonstrate the regulation of ecosystem
3. Illustrate process involved in transformation of natural calamity affected place to fertile land

Course Outcome 3 (CO3)

1. Demonstrate bio-geographical classification of biodiversity
2. Distinguish between in situ and ex situ conservation
3. Recall the term hot spots of biodiversity

Course Outcome 4 (CO4)

1. Compare the major limitations of the wildlife (protection) Act, 1972 and Forest (conservation) Act, 1980. Provide the effective ideas for the successful implementation of our environmental legislation.
2. Analyze the pollutants in the atmosphere are responsible for green house effect of Earth.
3. Differentiate between recycling and reuse**.**

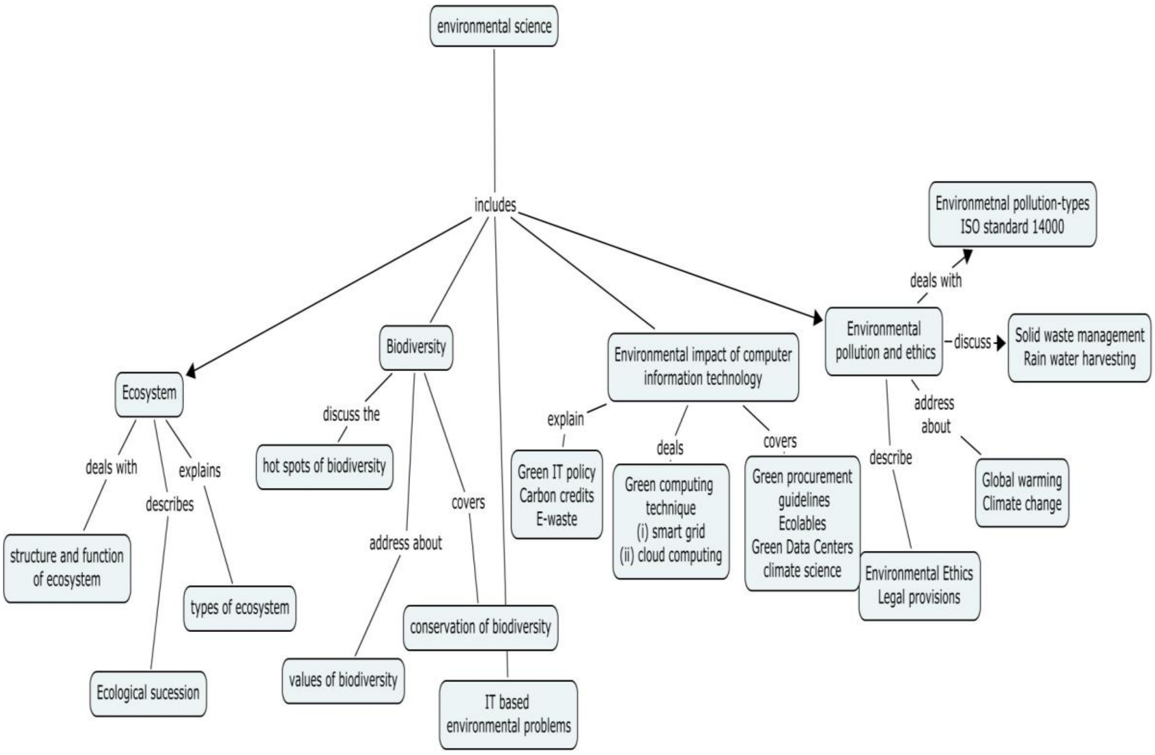
Course Outcome 5 (CO5)

1. India faces problems repeatedly due to flood in the rainy seasons. Account the problem and suggest suitable remedial measures.
2. While water is generally collected, stored and conserved at the surface in the form dams, lakes, ponds etc. Why is it essential to go for rain water harvesting by individuals specifically in large and thickly populated cities?
3. Outline the term sustainable development

Course Outcome 6 (CO6)

1. Highlight the areas of ICT that are directly contributing to organizations GHG emissions.
2. Explain the role of green disk in computer related wastes
3. List out the green procurement guidelines for the purchase of a personal compute

Concept Map



Syllabus

Ecosystem: Multidisciplinary nature of environment- need for public awareness-Eco-system-Concept, structure, function, components, laws of Ecology, Energy flow in eco system - Food chains, food webs-Ecological pyramids-Ecological succession. Types of eco system-Loss of ecosystem and its estimation. **Biodiversity:** Biodiversity and its types, bio-geographical classification, Values of biodiversity - Hot spots of biodiversity-threats to biodiversity-Biodiversity Indices-Endangered and endemic species- conservation of bio-diversity, Natural resources-Types and their uses-over exploitation. Conservation. **Environmental impact of computer Information Technology:** Role and Importance of Green IT policy, Dangers of Green wash, Carbon Footprint Calculators, Carbon Offsetting and Carbon Neutrality, Carbon trading, Techno trash, (E-Wastes) Green disk-its management, Green computing technology, Thin clients, Virtualization, Smart Grids, Cloud computing, Computational Energy Consumption, sustainable Green procurement guidelines, Ecolables. Green Data centers, Climate Science, Geomatics**. Environmental Pollution and Ethics:** Environmental pollution- types, effects and control measures – ISO 14000 standards, solid waste management–causes, effects and control measures. Water conservation - Rainwater Harvesting-Global warming-climate change and its effect on Environment – acid rain - ozone layer depletion-Environmental Ethics - sustainable development - Future aspects - Human and Animal rights-conservation of ethics and traditional value systems of India - Legal provisions-Environmental acts.

Reference Books

1. Anubha Kaushik and C.P. Kaushik, Environmental science and engineering, third edition, New age international (p) ltd publishers.
2. Mark G O' Neill, Green IT for sustainable Business Practice, An ISBN Foundation Guide.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Ecosystem**	
1.1	Multidisciplinary nature of environment-need for public awareness	1
1.2	Eco-system-Concept, structure, function, components	1
1.3	Laws of Ecology and Energy flow in eco system	1
1.4	Food chains, food webs-Ecological pyramids	2
1.5	Ecological succession and regulation	1
1.6	Types of ecosystem, and their Loss and estimation	2
2.	**Biodiversity**	
2.1	Types of biodiversity and their bio-geographical classification	1
2.3	Hot spots of biodiversity and biodiversity indices	1
2.4	Threats to biodiversity	1
2.5	Values of biodiversity	1
2.6	Endangered and endemic species of india	2
2.7	Conservation of biodiversity	2
3	**Environmental impact of computer Information Technology**	
3.1	Role and Importance of Green IT policy, Dangers of Green wash,	1
3.2	Carbon Footprint Calculators, Carbon Offsetting and Carbon Neutrality, Carbon trading	1
3.3	Techno trash, (E-Wastes) Green disk-its management	1
3.4	Green computing technology, Thin clients, Virtualization,	1
3.5	Smart Grids, Cloud computing, Computational Energy Consumption,	2
3.6	sustainable Green procurement guidelines, Ecolables.	1
3.7	Green Data centers, Climate Science, Geomatics.	2
4	**Environmental Pollution and Ethics:**	
4.1	Environmental pollution- types, effects	2
4.2	control measures – ISO 14000 standards,	2
4.3	solid waste management–causes, effects and control measures	1
4.4	Water conservation - Rainwater Harvesting-Global warming-climate change and its effect on Environment – acid rain - ozone layer depletion	3
4.5	Environmental Ethics - sustainable	2

Module No.	Topic	No. of Lectures
	development - Future aspects - Human and Animal rights-conservation of ethics and traditional value systems of India	
4.6	Legal provisions-Environmental acts.	1
	Total	**36**

Course Designers:

1. Dr.K.Radha hodchem@tce.edu
2. Mrs.J.Shanmugapriya shanmugapriya@tce.edu
3 Dr.S.Sivailango ssilango@tce.edu

14CS270	PROBLEM SOLVING USING COMPUTERS	Category	L	T	P	Credit
		ES	2	0	1	3

Preamble

This syllabus is intended for the candidate who desires to learn problem-solving techniques and the design of computer solutions in a precise manner. The syllabus emphasizes problem-solving methodologies, algorithm designs and developments and computer-programming skills. The intention is to provide sufficient depth in these topics to enable candidates to achieve better understanding of problem solving using computers. Besides the written papers, lab-based examinations are included as part of the assessment requirements for the study. The lab-based examinations will test the candidate's ability to develop computer-programming solutions for a series of programming tasks of varying complexity.

The modules in the syllabus reflect solving general problems via programming solution. Thus, modules collectively focus on programming concepts, strategies and techniques; and the application of these toward the development of programming solutions.

Course Outcomes

On the successful completion of the course, students will be able to

Develop algorithms for solving simple mathematical and engineering problems and examine the suitability of appropriate repetition and/or selection structures for given problems (CO1)	Apply
Solve matrix problems, merging, searching, sorting and string manipulation problems using iteration, modularization or recursion as applicable. (CO2)	Apply
Organize files to perform text operations like editing, pattern searching using structures. (CO3)	Apply
Implement the algorithms for matrix problems, merging, searching, sorting, and string manipulation and file problems and debug and test using any procedural programming language (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	S	M						M			M
CO2.	S	S	M						M			M
CO3.	S	S	M						M			M
CO4.	S	S	M						M			M

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1(T)	2(T)	3(Practical)	Theory
Remember	20	20	-	20
Understand	20	20	-	20
Apply	60	60	100	60
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Recall the list of symbols used in flowcharts for various purposes. (Remember)
2. Summarize the steps involved in exchanging values of variables. (Understand)
3. Choose proper selection control structures to solve area of rectangle, triangle and circle. (Apply)

Course Outcome 2 (CO2):

1. What is the use of an array? (Remember)
2. Compare function call and recursive call. (Understand)
3. Make use of arrays and functions to transpose an mxn matrix. (Apply)
4. Analyze the performance of search algorithms. (Analyze)

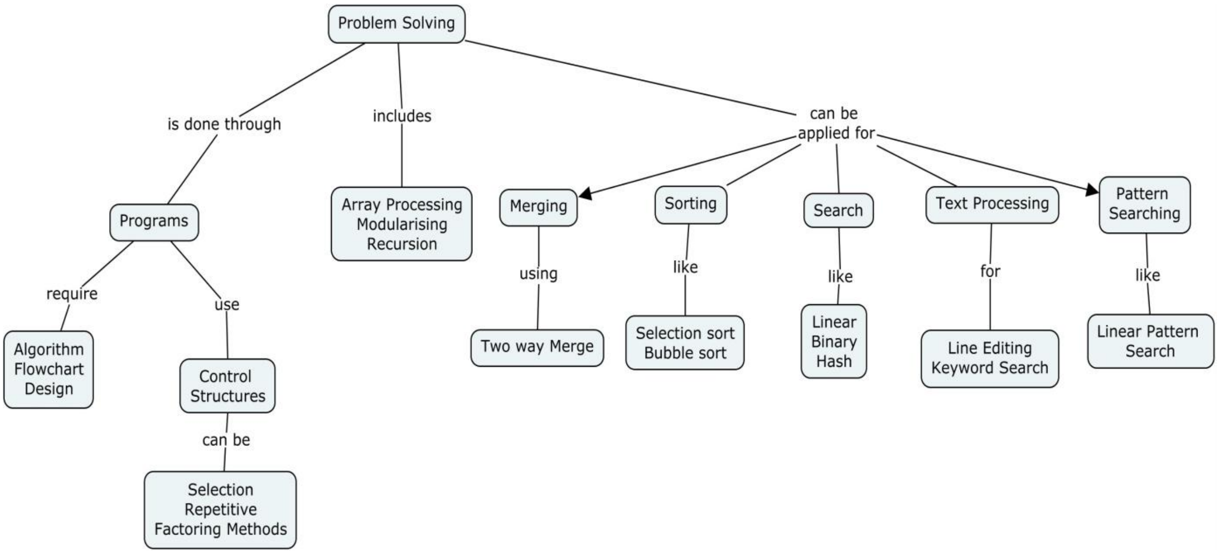
Course Outcome 3 (CO3):

1. What is text processing?
2. Explain the algorithm for linear pattern searching
3. Develop an algorithm for comparing two strings.

Course Outcome 4 (CO4):

1. Develop a C program to convert decimal to binary of a given number using non recursive and recursive techniques. (Apply)
2. Develop a C program to multiply two nxn matrices using arrays and pointers. (Apply)
3. Develop a C program to create a text file to store records of addresses of N persons and retrieve and display the records with city="Madurai". (Apply)

Concept Map



Syllabus

Introduction to Computer Problem Solving, Program Design, Flowcharts, developing an Algorithm, Efficiency of algorithms, Analysis of algorithms, Fundamentals Algorithms

Practical Component
Problem Solving with Fundamental Algorithms (use data types and expressions)

Selection Control Structures, Repetition Control Structures, Algorithms Using Selection and Repetition, Factoring Methods

Practical Component
Problem solving with Selection Control Structures and Decision Statements (use if-else, switch-case, break, and continue)
Problem solving with Repetition Control Structures and Loop Statements (use while, do-while and for loops)

Array Processing and Techniques, Modularization and recursion, Merging, Sorting and Searching- Two way merge, Sorting by selection, Linear search, Binary search, Simple Hash searching
Practical Component
Problem solving with array based problems (use 1D and 2D arrays and pointers) and function oriented problems (functions and recursive functions)

Text Processing and pattern searching, Text line editing, keyword searching, linear pattern searching
Practical Component
Problem solving using text and strings (use string, structures and files**)**

Text Books

1. How to solve it by Computer, R.G Dromey, Pearson education, Delhi, 2008.
2. Simple Program Design, A Step-by-Step Approach, Lesley Anne Robertson, 5th Edition, Thomson, 2007.

Reference Books

1. Let us C, Yashavant P. Kanetkar ,12th edition, BPB Publications, 2012.
2. ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/download-course-materials/

Course Contents and Lecture Schedule for Theory

Module No.	**Topic**	**No. of Lectures**
1	Introduction to Computer Problem Solving	
1.1	Introduction to Computer, Program Design, Developing an Algorithm	1
1.2	Flowcharts	1
1.3	Efficiency of algorithms, Analysis of algorithms	1
1.4	Fundamentals Algorithms	1
	Exchanging values of variables, Counting	1
2	Factoring methods and Control structures	
2.1	Selection Control Structures, Repetition Control Structures	1
2.1.1	Summation of set of numbers, Factorial computation, Sine function computation	1
2.1.2	Fibonacci sequence generation, Reversing the digits of an Integer	1
2.1.3	Base conversion, Character to number conversion	1
2.2	Factoring Methods	
2.2.1	Finding square root of a number, The smallest divisor of an integer	1
2.2.2	Generating Prime numbers	1
2.2.3	Generating Pseudo-random numbers, Computing nth Fibonacci number	1
3	Array Processing and Techniques	

Module No.	Topic	No. of Lectures
3.1.1	Array technique, Finding the maximum number in a set, Finding k^{th} smallest number	1
3.1.2	Removal of duplicates from an ordered array, Partitioning array,	1
3.1.3	Histogramming, Longest Monotone subsequence	1
3.2	Matrix manipulations (add, subtract, multiply, transpose)	2
3.3	Modularization and recursion, Sorting by selection, Two way merge	1
3.4	Linear search, Binary search, Simple Hash searching	2
4	Text Processing and pattern searching	
4.1	String Manipulations	1
4.2	Text line editing	1
4.2.1	keyword searching	1
4.2.2	linear pattern searching	1
	Total	**24**

Course Contents and Lecture Schedule for Laboratory

Module No.	Topic	No. of Lectures
1	Introduction to C components	1
2	Problems on Fundamentals Algorithms	2
3	Factoring Methods in C	1
4	Problems on Factoring Methods	2
5	Selection Control Structures, Repetition Control Structures in C	1
6	Problems on Selection Control Structures, Repetition Control Structures	2
7	Array techniques in C	1
8	Problems on Array techniques	2
9	Functions and recursion in C	1
10	Problems on Functions and recursion	2
11	Concepts of String in C	1
12	Problems on Strings	2
13	Concepts of structures in C	1
14	Problems on structures	2
15	Concepts of Files in C	1
16	Problems on Files	2
	Total	**24**

Course Designers:

1. Mrs. S. Sudha ssj@tce.edu
2. Mr. S. Prasanna sprcse@tce.edu

14CS280	DIGITAL CIRCUITS LABORATORY	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

The laboratory course is designed to enable the students to design and construct practically the combinational and sequential logic circuits for different applications. The list of experiments starts with the verification of Boolean theorems and truth table of gates. Then the design and construction of a variety of circuits using gates, flip flops and other devices are performed. The simulation of simple circuits using Hardware Description Language is also performed. These experiments will reinforce the concepts learnt in the corresponding theory course.

Prerequisite

14ES160: Basics of Electrical and Electronics Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Build combinational logic circuits for a given application using logic gates, multiplexers, decoders and encoders. (CO1)	Apply
Build sequential logic circuits for a given application using the given type of flip flops. (CO2)	Apply
Simulate and test simple combinational logic circuits using Hardware Description Language (HDL). (CO3)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	L	S							L		
CO2.	S	L	S							L		
CO3	S	L	S							L		

S- Strong; M-Medium; L-Low

List of Experiments

1. Verification of truth tables of logic gates and theorems of Boolean algebra.
2. Design of half adder, full adder and parallel binary adder.
3. Design of BCD adder
4. Design of 8 to 1 multiplexer and make use of it to implement a full adder.
5. Design of decimal to binary encoder.
6. Design of 2-bit magnitude converter.
7. Design of binary to excess 3 code converter
8. Design of BCD to decimal decoder
9. Design of 4-bit ripple up and down counters
10. Design of 2-bit synchronous counters to count in a specified sequence.
11. Design of 4-bit shift register, ring counter and Johnson counter.
12. Simulation of a 4-bit parallel binary adder using HDL .
13. Simulation of a multiplexer and decoder using HDL.

Course Designers:

1.	Mr. C.Sridharan	cscse@tce.edu
2.	Dr. N.Balamurugan	nbbalamurugan@tce.edu

			Category	L	T	P	Credit
14CS290	**WORKSHOP**		ES	0	0	1	1

Preamble

This is the foundation practical course for the students of circuit branches (EEE, ECE, CSE and IT). The aim of this course is to impart fundamental hands-on skill in carrying out experiments at higher semester practical courses.

Prerequisite

14ES160 : Basic Electrical and Electronics Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Practice on different Unix and DOS commands. (CO1)	Apply
Prepare configuration management of Windows operating system. (CO2)	Apply
Practice on designing and preparing reports using word, Power-point and Excel applications (CO3)	Apply
Review on Rapid prototyping tools. (CO4)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M			L							L
CO2	S	M	S		L							L
CO3	S	M	S		L							L
CO4	L	L	L	L	L							L

List of Experiments

EEE:

1. Realization and Discrimination of fuses and Circuit breakers
2. Earthling practices and its significances
3. Wiring practices and testing
4. Functionalities of RPS/AFO/CRO
5. Functionalities and Selection of Analog and Digital meters

ECE:

1. Identifying electronic components and understanding PCB glossary
2. Conversion of schematic into PCB layout and PCB fabrication
3. Practicing of soldering and desoldering

Computer Science and Engineering:

1. Practice on different DOS and Unix commands. Basic configuration management of Windows operating system.
2. Practice on designing and preparing reports using word, Power-point and Excel applications.
3. Study on Rapid prototyping tools

Information Technology:

1. **Computer Assembly and Configuration: PC Assembling:** Steps for assembling a PC-commonly used devices an overview, assembling a SMPS in a cabinet, fixing a processor in a mother board, assembling RAM in a motherboard, pinning a cooling fan in a mother board, Assembling a hard disc drive in a cabinet, assembling a CD/DVD ROM in a cabinet. Assembling a floppy drive in a cabinet, fixing motherboard In a cabinet, Connecting the cables from the SMPS to motherboard, hard disc, drives & etc, Establishing data connection for to motherboard, hard disc, drives. Fixing wires for power restart switches, fixing wires for power & HDD LED's, fixing wires for external USB and Audio connections.
2. **System Installation:** Steps for installing software's for hardware, Hardware & Software Trouble Shooting.

Course Designers

1.	Dr.V.Saravanan	vseee@tce.edu
2.	Dr.V.Prakash	vpeee@tce.edu
3.	Dr.P.S.Manoharan	psmeee@tce.edu
4.	Dr.K.Hariharan	khh@tce.edu
5.	Mr. M.Sivakumar	mskcse@tce.edu
6.	Mr.C.Senthilkumar	cskcse@tce.edu
7.	Mr. M.Thangavel	thangavelmuruganme@gmail.com

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

THIRD SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING

(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY
(For the candidates admitted from 2016-17 onwards)

THIRD SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS310	Probability and Statistics	BS	2	2	-	3
14CS320	Theory and Design of Programming Languages	PC	3	-	-	3
14CS330	Computer Graphics	PC	3	-	-	3
14CS341	Engineering Design	ES	1	-	2*	3
14CS350	Data Structures and Algorithms	PC	3	-	-	3
THEORY CUM PRACTICAL						
14CS370	Object Oriented Programming	PC	2	-	2	3
PRACTICAL						
14CS380	Data Structures Lab	PC	-	-	2	1
14CS390	Assembly Language Programming Lab	PC	-	-	2	1
Total			**14**	**2**	**8**	**20**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core
L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture is equivalent to 1 credit

2 Hours Tutorial is equivalent to 1 credit

2 Hours Practical is equivalent to 1 credit

* - 2 hours/ week is allotted for off-class practical work

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

THIRD SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS310	Probability and Statistics	3	50	50	100	25	50
2	14CS320	Theory and Design of Programming Languages	3	50	50	100	25	50
3	14CS330	Computer Graphics	3	50	50	100	25	50
4	14CS341	Engineering Design	-	100	-	100	-	50
5	14CS350	Data Structures and Algorithms	3	50	50	100	25	50
THEORY CUM PRACTICAL								
7	14CS370	Object Oriented Programming	3	50	50	100	25	50
PRACTICAL								
8	14CS380	Data Structures Lab	3	50	50	100	25	50
9	14CS390	Assembly Language Programming Lab	3	50	50	100	25	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

		Category	L	T	P	Credit
14CS310	**PROBABILITY AND STATISTICS**	BS	2	1	0	3

Preamble

An engineering student needs to have some basic mathematical tools and techniques to apply in diverse applications in Engineering. This emphasizes the development of rigorous logical thinking and analytical skills of the student and appraises him the complete procedure for solving different kinds of problems that occur in engineering. Based on this, the course aims at giving adequate exposure in random variables, probability distributions, regression and correlation, test of hypothesis and statistical quality control.

Prerequisite

14MA110 Engineering Mathematics

Course Outcomes

On the successful completion of the course, students will be able to

Infer expectation, variance, standard deviation moments and moment generating function for discrete and continuous random variables (CO1)	Understand
Apply the concept of expectation and moment generating functions to discrete and continuous distributions and find the probability values for the defined distributions (CO2)	Apply
Apply the concept of linear and non linear correlation, linear and non linear regressions to engineering problems (CO3)	Apply
Apply least square method in fitting linear and non linear regression curves. (CO4)	Apply
Demonstrate the concept of testing of hypothesis for small and large samples by using various tests like t-test, F-test, z-test and chi-square test (CO5)	Understand
Identify whether two samples came from same population or from different population for a set of sample data (CO6)	Understand
Apply statistical principles for the quality control of manufacturing processes (CO7)	Apply

Mapping with Program Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	P012
CO1	L											
CO2	M											
CO3	S	L										
CO4	S	L										
CO5	S	L										

CO6	S	L										
CO7	M											

Correlation: S-Strong; M-Medium; - L- Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	10	10	10	0
Understand	30	30	30	30
Apply	60	60	60	70
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe Continuous random variable with an example.
2. Define probability mass function of hyper geometric distribution.
3. Estimate the moment generating function of the random variable X given pdf

   f(x) = $2e^{-2x}$; x > 0

Course Outcome 2 (CO2):

1. Predict the value of 'a' if P(X = x) = a $(2/3)^x$; x = 1,2,3……
2. The distribution function of a Random variable X is given by F(x) = 1 – (1+ x) e^{-x} ; x ≥ 0.

   Find the density function and mean.
3. If the probability that an applicant for a driver's license will pass the road test on any given trial is 0.8. What is the probability that he will finally pass the test (a) on the fourth trail and (b) in fewer than 4 trails?

Course Outcome 3 (CO3):

1. Compute $R_{1.23}$ if r_{12}=0.77; r_{13}=0.72; r_{23}=0.52.
2. Differentiate between correlation and regression of variables
3. Coefficient of correlation between x and y is 0.48. Their covariance is 36. The variance is 16. Find the standard deviation of y.

Course Outcome 4 (CO4):

1. Fit a parabola for a following data

X: 1	2	3	4	5	6	7	8	9	10
Y: 2	6	7	8	10	11	11	10	9	7

2. Determine the plan of regression of Y on X1 and X2 for the following data:

Y :	90	72	54	42	30	12
X1:	3	5	6	8	12	14
X2:	16	10	7	4	3	2

3. In a distribution $\sigma_1 = 2, \sigma_2 = 3, \sigma_3 = 3, r_{12} = 0.7, r_{23} = 0.5, r_{31} = 0.5$ Find

   (i) $r_{23.1} (ii) R_{1.23} (iii) b_{12.3}, b_{13.2} (iv) \sigma_{1.23}$

Course Outcome 5 (CO 5):

1. Explain Null hypothesis.
2. State the important properties of'tʼ distributions.
3. Twenty people were attacked by a disease and only 18 survived. Will you reject the hypothesis that the survival rate if attacked by the disease is 85% in favor to the hypothesis that it is more at 5% level?

Course Outcome 6 (CO 6):

1. The nicotine contents in two samples of tobacco are given below:

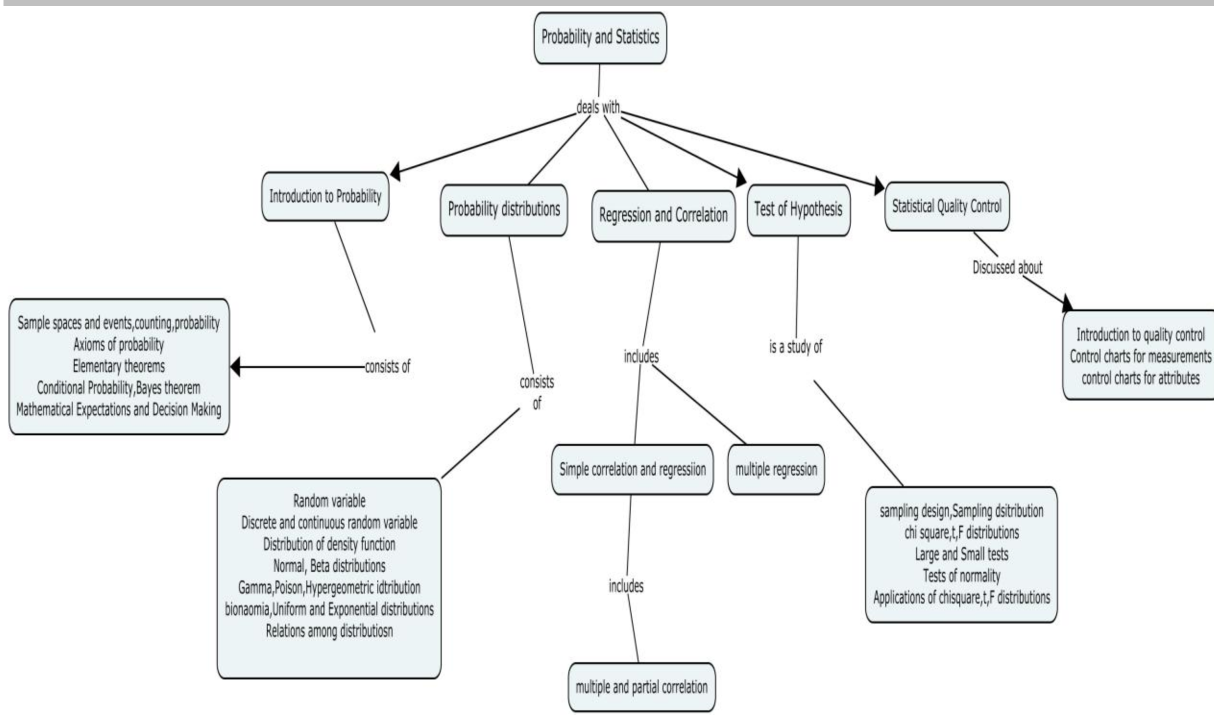
Sample 1:	21	24	25	26	27	-
Sample 2:	22	27	28	30	31	36

   Can you say that the two samples came from the same normal population?
2. In a random sample of size 500 the mean is found to be 20. In another sample of size 400 the mean is 15. Could the samples have been drawn from the same population with SD 4?
3. The average marks scored by 32 boy's is72 with a S.D. of 8, while that for 36 girls is 70 with a S.D. of 6. Test at 1% level of significance whether the boys perform better than girls.

Course Outcome 7 (CO 7):

1. 10 samples each of size 50 were inspected and the numbers of defectives in the inspection were 2,1,1,2,3,5,5,1,2,3.Construct the appropriate control chart for defectives.
2. Distinguish between p-chart and c-chart.
3. Fifteen samples each of size 50 were inspected and the number of defectives in the inspection were:
   2,3,4,2,3,0,1,2,2,3,5,5,1,2,3. Draw the control chart for the number of defectives and comment on the state of control.

Concept Map



Syllabus

Introduction to Probability: Sample spaces and events, counting, probability, the Axioms of probability, Some elementary theorems, Conditional probability, Baye's theorem, Mathematical Expectation and Decision Making. **Probability distributions:** Random variables, Discrete, continuous Random variables, Distribution and density functions, Normal, Beta, Gamma, Poisson, Hyper geometric, Binomial, Uniform and Exponential distributions, and some inter-relationships among the various distributions. **Regression and Correlation:** Simple linear correlation and regression, multiple regression analysis, multiple and partial Correlation

Coefficients. **Test of Hypothesis:** Sampling design, sampling distributions, chi-square, 't', 'F' distribution, Large and small sample tests, Test for (1)Proportion (2) Mean (3)Variance and (4) Difference between two proportions, Means and variances in large and small samples, Tests of normality, Applications of chi-square, 't', 'F' distributions for test of hypothesis. **Statistical Quality Control:** Introduction to quality control, control charts for measurements, control charts for attributes. Demonstrate the above control charts and testing methods through Mathematical software- SYSTAT.

Text Books

1. S.C.Gupta, V.K.Kapoor, " Fundamentals of Mathematical Statistics", Eighth Edition, Sultan Chand and Sons, New Delhi, 2001
2. Michael Baron, "Probability and Statistics for Computer Scientists "second edition,CRC press,USA.
3. Veerarajan.T, " Probablility and Statistics" Tata McGraw-Hill Limited, New Delhi

Reference Books

1. R.C. Saxena, J.N. Kapoor, "Mathematical Statistics", S.Chand and Co, 1999.
2. J.N.Sharma, J.K.Goel, "Mathematical Statistics", Seventh Edition, Krishna Prakasham Mandir, Meerut, 1998.
3. Miller, Fan, "Probability and Statistics for Engineers", Prentice Hall of India, 2001.
4. Jay L Devore, "Probability and Statistics for Engineering and the Sciences", Thomson Asia, 2002.

Course Contents and Lecture Schedule

S.No	Topics	No. of Lectures
1	**Introduction to probability**	
1.1	Sample spaces and events, counting, probability, the Axioms of probability	1
1.2	Some elementary theorems, Conditional probability	1
	Tutorial	2
1.3	Baye's theorem, Mathematical Expectation and Decision Making	1
	Tutorial	3
2	**Probability distributions**	
2.1	Random Variables, Discrete and Continuous RVs	1
2.2	Distribution and Density functions	1
2.3	Normal, Beta distributions	1
2.4	Gamma, Poisson Distributions	1
	Tutorial	3

2.5	Hyper Geometric Distributions	1
2.6	Binomial, Uniform and Exponential distributions	2
	Tutorial	2
3	**Regression and Correlation**	
3.1	Simple linear correlation and regression	1
	Tutorial	2
3.2	Multiple regression	1
3.3	multiple and partial Correlation Coefficients	1
	Tutorial	3
4	**Test of Hypothesis**	
4.1	Sampling design and Sampling Distributions	1
4.2	Chi-square, t, F distributions	2
4.3	Large Sample Tests- Z test for proportion, mean	1
	Tutorial	3
4.5	Small sample Tests- t test, F test	2
4.6	Chi-square test, Test of Normality	1
4.7	Application of Various Tests	1
	Tutorial	3
5	**Statistical Quality Control**	
5.1	Introduction to quality control ,control charts for measurements	1
5.2	control charts for attributes	1
5.3	Demonstrate the above control charts and testing methods through Mathematical software- SYSTAT.	1
	Tutorial	3

Course Designers

1. Mr. N.K.Chandrasekaran nkcmat@tce.edu
2. Mr. B.Vigneswaran bvmat@tce.edu
3. Dr.A.Anitha anithavalli@tce.edu

14CS320	THEORY AND DESIGN OF PROGRAMMING LANGUAGAES	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

The course will enable the student to understand the concepts of different programming paradigms like imperative programming, Object Oriented programming, Logic programming, Functional programming and Concurrent programming.

Prerequisite

14CS270: Problem Solving using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Describe the programming language basics and language processing activities. (CO1)	Understand
Explain the syntax of a programming language using Context-Free Grammars.(CO2)	Understand
Construct code-snippets for solving simple engineering problems like searching, list manipulations with an understanding of appropriate language constructs.(CO3)	Apply
Infer the programming paradigm to which a given programming language belongs. (CO4)	Understand
Identify a suitable programming paradigm for the given application (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M		L									
CO2	M		L									
CO3	L	L										
CO4	S	M	M									
CO5	S	M	M		L	L			M	M		

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	60	50	40	40
Apply	20	30	40	40

Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List the characteristics of high level language.
2. Discuss any two special cases in loop construct.
3. Differentiate between compiler and interpreter.

Course Outcome 2 (CO2):

1. Write the prefix and postfix notation for the given expression.
2. Draw flow diagrams for the following program fragment.

```
if E1 then S1
 else if E2 then S2
        else S3
end
```

3. Define Context-Free Grammar.

Course Outcome 3 (CO3)

1. Explain the two syntactic concerns in Pascal by comparing with Modula-2.
2. Illustrate Lexical scope and Dynamic scope.
3. Illustrate the various OOPS features.

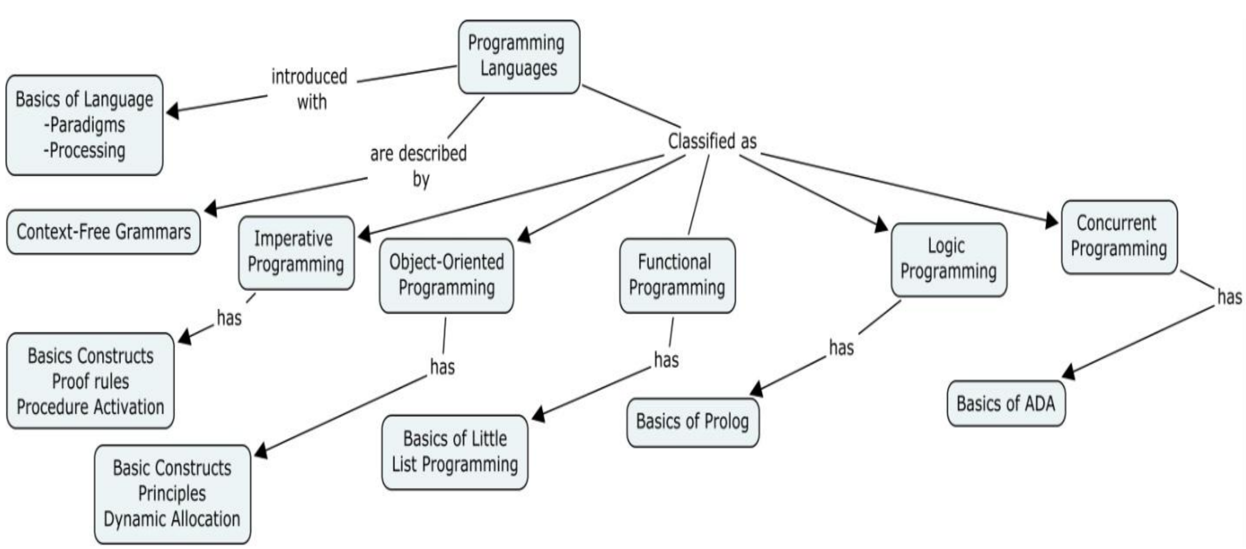
Course Outcome 4 (CO4)

1. Distinguish the two basic approaches to implement a program in high level language.
2. Review three different constructs for program structuring with a common application as example.
3. Explain the use Exported and Imported names.

Course Outcome 5 (CO5)

1. Compare dangling pointers and memory leaks.
2. Write a Pascal program to create two lists L1 and L2 of same type using pointer and record data types. Generate a new list L3 by performing L1 U L2 (Union Operation).
3. Choose the best programming paradigm for a Transaction Processing system.

Concept Map



Syllabus

Basics of Programming Languages: Introduction to programming languages, Programming Paradigms, Language Processing Activities, Fundamentals of Language Processing **Language Description:** Expression Notations, Abstract Syntax Trees Lexical Syntax, Context-Free Grammars, Grammars for Expression **Imperative Programming:** Syntax-Directed Control Flow, Programming with Invariants, Proof rules for Partial Correctness, Basic Types, Arrays & Records, Unions & Sets, Pointers, Procedure Activations, Imperative Programming Languages **Object-Oriented Programming:** Constructs for Program Structuring, Object Model, parameter passing, Dynamic Allocation, Object-Oriented Programming Languages **Functional Programming:** A Little Language of Expressions, Types and Expression Evaluation, Functional Programming with Lists **Other Paradigms:** Introduction to Logic Programming, Basics of Prolog, Introduction to Concurrent Programming, Concurrency in ADA. **Haskell, Ruby**

Text Book

1. Ravi Sethi, "Programming Languages: Concepts and Constructs", AT&T Bell Laboratories, 2nd edition, Addison Wesley,2007.

Reference Books

1. D.M.Dhamdhere, "System Programming", Tata McGraw Hill, 2011.
2. Allen B.Tucker, Robert E.Noonan, "Programming Languages Principles and Paradigms" 2nd Edition, Tata McGraw Hill, 2007.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Basics of Programming Languages**	
1.1	Introduction to programming languages	1
1.2	Programming Paradigms	1
1.3	Language Processing Activities	1
1.4	Fundamentals of Language Processing	1
2	**Language Description**	
2.1	Expression Notations, Abstract Syntax Trees	1
2.2	Lexical Syntax, Context-Free Grammars	1
2.3	Grammars for Expression	
3	**Imperative Programming**	
3.1	Syntax-Directed Control Flow	1
3.2	Programming with Invariants	1
3.3	Proof rules for Partial Correctness	1
3.4	Basic Types	1
3.5	Arrays & Records	1
3.6	Unions & Sets	1
3.7	Pointers	1
3.8	Procedure Activations	1
3.9	Imperative Programming Languages	2
4	**Object-Oriented Programming**	
4.1	Constructs for Program Structuring	2

Module No.	Topic	No. of Lectures
4.2	Object Model	2
4.3	Parameter passing	2
4.4	Dynamic Allocation	1
4.5	Object-Oriented Programming Languages	1
5	**Functional Programming**	
5.1	A Little Language of Expressions	2
5.2	Types and Expression Evaluation	2
5.3	Functional Programming with Lists	2
6	**Other Paradigms**	
6.1	Introduction to Logic Programming	1
6.2	Basics of Prolog	2
6.3	Introduction to Concurrent Programming	1
6.4	Concurrency in ADA	2

Course Designers:

1. Mrs. G.Madhu Priya gmadhupriya@tce.edu
2. Mrs. M.Suguna mscse@tce.edu
3. Mr. S.Karthick skcse@tce.edu

14CS330	COMPUTER GRAPHICS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This course will cover 2D and 3D graphics. Students will learn how to transform and represent 2D and 3D objects. At the end of the course students will shine in animation field.

Prerequisite

14CS270 : Problem Solving using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Illustrate the working of appropriate drawing and clipping algorithms for 2D objects. (CO1)	Apply
Produce an object after applying the required 2D/ 3D transformation techniques. (CO2)	Apply
Explain different color models like RGB and CMYK. (CO3)	Understand
Identify the visible and invisible surfaces of 3D objects by applying a suitable surface detection algorithm. (CO4)	Apply
Develop 2D/3D animation for a given scenario by applying the principles of animation. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L									
CO2	S	M	L									
CO3	M	L	L									
CO4	S	M	L									
CO5	S	M	S					L	L	L		L

S- Strong; M-Medium; L-Low

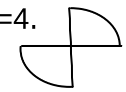
Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	20	30	20	20
Apply	50	50	60	60
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Use midpoint ellipse algorithm to determine the pixel values for an object shown below. An object is centered at (-2, 3) and having ellipse parameters r_x=3 and r_y=4.



2. Use Bresenham line drawing algorithm to find the intermediate pixel values for the following lines
   Line 1: A (7, 8) & B (2, 4)   Line 2: C (15, 25) & D (10, 31)

3. Differentiate Parallel line from DDA line algorithm? Illustrate the derivation and hence obtain a procedure to draw a line using DDA line drawing algorithm.

4. Compute the output of Liang Barsky Line clipping algorithm to clip the line P1P2

   Where P1=(22,47) and P2=(62,67) and the window size is given by (30,40,80,70).
5. Explain the procedure of curve clipping using circle.

Course Outcome 2 (CO2):

1. Estimate the new co-ordinate position of a triangle: A (1, 1), B (1, 6) and C (4, 1) after reflection i) through perpendicular of xy plane ii) about the line y =6x+5
2. Illustrate the effect of shearing transformations on the square A (0, 0), B (4, 0), C (4, 4) and D (0, 4)
   a) X direction shearing when sh_x= - 6
   b) Y direction shearing with respect to x_{ref} =3 and sh_y=3
   c) Shearing in both directions when sh_x= 5 and sh_y= 5
   Distinguish coordinate from geometric transformation.
3. Determine the new co-ordinate position on the following points A(4,3,8) , B(-7,5,8) and C(6, 8,-4) after scaling w.r.to pivot point (3,4,3) and the scaling factor is given by (8,8,5).
4. Perform 3D rotation on a unit cube about y axis w.r.to pivot point (4, 0,-7) and θ =60 degree.
5. Distinguish Co-ordinate from Geometric transformation.

Course Outcome 3 (CO3):

1. Compare and Contrast RGB, YIQ and CMY Model.
2. Explain the conversion Between HSV and RGB Models.
3. Give the transformation matrix for the conversion of RGB to CMY color model.

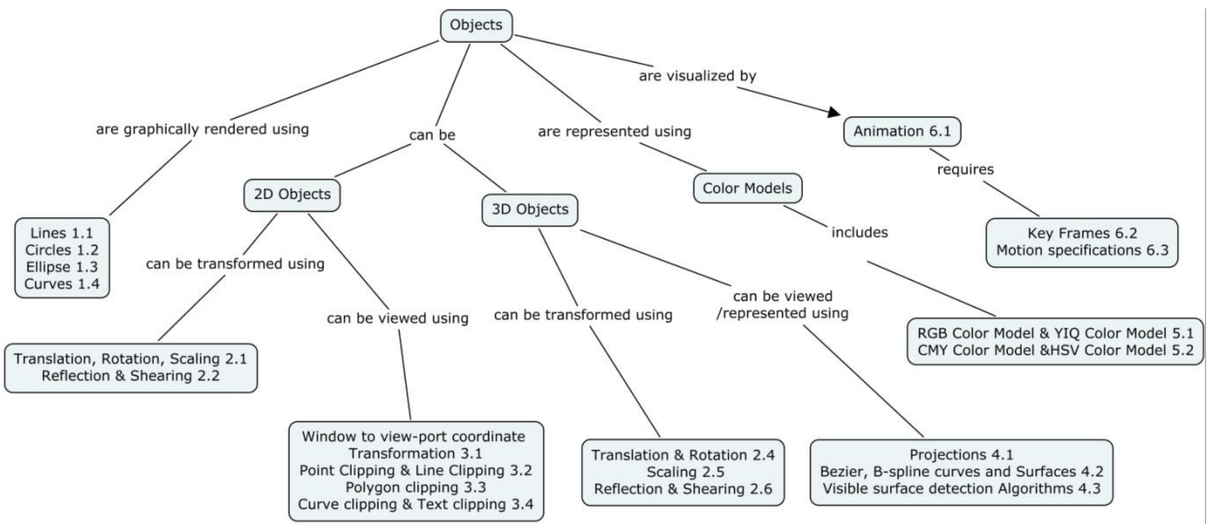
Course Outcome 4 (CO4)

1. Explain how depth buffer method can be used to find out visible surfaces in a scene?.
2. Explain how scan line and back face detection algorithms can be used to find out visible surfaces in a scene?.
3. Difference between object and image space method.

Course Outcome 5 (CO5)

1. Explain how to create a running tiger from moving automobile using morphing effect.
2. Explain how to create a bouncing ball using different types of motion specifications.
3. Discuss about principles of animation.

Concept Map



Syllabus

Introduction to Computer Graphics : Graphics types and its applications, Points and Lines, Line Drawing Algorithms- DDA, Bresenham and Parallel Line Algorithm, Midpoint Circle drawing Algorithm, Ellipse generating Algorithm, Parallel Curve Algorithm.

2D and 3D Geometric Transformations: Basic Transformations- Translation, Rotation, Scaling, Reflection, Shearing, Composite Transformations

2D Viewing: Viewing Pipeline, Window to view-port coordinate Transformation, Clipping Operations- Point Clipping, Cohen Sutherland Line Clipping, Liang Barsky Line Clipping, Sutherland Hodgeman Polygon Clipping, Weiler - Atherton Polygon Clipping, Curve and Text Clipping.

3D Viewing and Object Representation: Projections – Parallel and Perspective Projection, Bezier Curves and Surfaces, B-Spline Curves and Surfaces, Visible Surface Detection Algorithms- Back-Face Detection Algorithm, Depth Buffer Method, Scan line Method**.**

Color Models: RGB Color Model ,YIQ Color Model ,CMY Color Model, HSV Color Model, Conversion Between HSV and RGB Models and HLS Color Model

Animation: Principles of animation, Design of Animation Sequences , Key- Frame Systems : Morphing and Simulating Accelerations , Motion Specifications .

Text Book

1. Donald Hearn and M. Pauline Baker: Computer Graphics, PHI/Pearson Education, Second Edition, 2004.

Reference Books

1. F.S. Hill, Computer Graphics using OPENGL, Second edition, Pearson Education,2003.
2. James D. Foley, Andries Van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics- Principles and practice, Second Edition in C, Pearson Education, 2007.
3. Zhigang Xiang and Roy A.Plastock "Schaum's Outline of Computer Graphics" Second Edition, McGraw -Hill 2000.

Course Contents and Lecture Schedule

Module No.	Topic	No of Lectures
1.	**Rendering of Objects**	
1.1	Rendering of lines ((DDA Algorithm, Bresenham's Line Algorithm, and Parallel Line Algorithm)	2
1.2	Rendering of circles (Mid-point circle algorithm)	1
1.3	Rendering ellipses (Mid-point ellipse algorithm)	1
1.4	Rendering curves (Parallel curve algorithm)	1
2	**2D and 3D Geometric Transformations**	
2.1	2D Translation, Rotation and Scaling	1
2.2	2D Reflection and Shearing	2
2.3	Problems in 2D Transformations	2
2.4	3D Translation, Rotation , Scaling	2
2.5	3D Reflection and Shearing	1
2.6	Problems in 3D Transformations	2
3.	**2D Viewing**	
3.1	Viewing Pipeline, Window to view-port coordinate Transformation	1
3.2	Clipping - Point Clipping , Line Clipping Algorithms (Cohen Sutherland Line Clipping, Liang Barsky Line Clipping)	2
3.3	Sutherland Hodgeman Polygon Clipping , Weiler - Atherton Polygon Clipping	2

3.4	Curve and Text Clipping	1
3.5	Problems in Clipping	1
4	**3D Viewing and Object Representation**	
4.1	Projections – Parallel and Perspective Projection	2
4.2	Bezier, B-spline curves and Surfaces	3
4.3	Visible surface detection Algorithms (Back-Face Detection Algorithm , Depth Buffer Method, Scan line Method)	2
5	**Color Models**	
5.1	RGB Color Model ,YIQ Color Model ,	1
5.2	CMY Color Model, HSV Color Model,	1
5.3	Conversion Between HSV and RGB Models and HLS Color Model	2
6	**Animation**	
6.1	Principles of animation, Design of Animation Sequences	1
6.2	Key- Frame Systems : Morphing and Simulating Accelerations	1
6.3	Motion Specifications	1
	Total	**36**

Course Designers:

1. Mrs. S.Sridevi sridevi@tce.edu
2. Ms. G.Thiraviaselvi gts@tce.edu

14CS341	**ENGINEERING DESIGN**		Category	L	T	P	Credit
			ES	1	0	2*	3

Common for B.E./B.Tech Degree Programmes
(Course Codes: 14CE450, 14ME420, 14EE450, 14EC450, 14IT450, 14CS341,14MT420)

Preamble

Engineering design is normally taught, not as a unified course in India. The courses like Product design, Machine design, Electrical machine design and transformer design, Control system design and Communication system design are tailored to specific topics There were many new approaches developed over a period of time. There is a need to discuss a unified approach of design in a course.

Prerequisite

None

Course Outcomes

On the successful completion of the course, students will be able to

CO1: Explain the steps involved in Engineering Design	Understand
CO2. Explain the Engineering Design process and review designs with societal considerations.	Understand
CO3: Provide specification for customer needs/requirements, considering engineering Characteristics and quality Function Deployment.	Apply
CO4: Prepare conceptual design document.	Apply

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		
	CAT 1	Review 1	Review 2
Remember	20	0	0
Understand	40	0	0
Apply	40	100	50
Analyse	0	0	50
Evaluate	0	0	0
Create	0	0	0

- Milestones:
  1. Problem description (3 weeks)
  2. Framework (4 weeks)
     i. Functional requirements
     ii. User requirements
     iii. Performance requirements
     iv. Specifications
  3. Preliminary design (conceptual) (3 weeks)
     i. Cost estimates
  4. Final design (conceptual document) (2 weeks)

Review 1 for milestones1 & 2 and Review 2 for milestones 3 & 4

* - 2 hours/ week is allotted for off-class practical work

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define Engineering Design
2. State different activities involved in Product Engineering Life Cycle
3. List different design considerations that are required for a good design
4. Explain different types of design
5. List the characteristics of environmentally responsible design

Course Outcome 2 (CO2):

1. List different modes to collect user requirements.
2. Briefly explain the classification of different types of User requirement
3. Define Benchmarking or Reverse Engineering or Product Dissection
4. List two categories of Redesign
5. Explain different activities involved in Design process
6. Explain different steps involved in Conceptual Design process

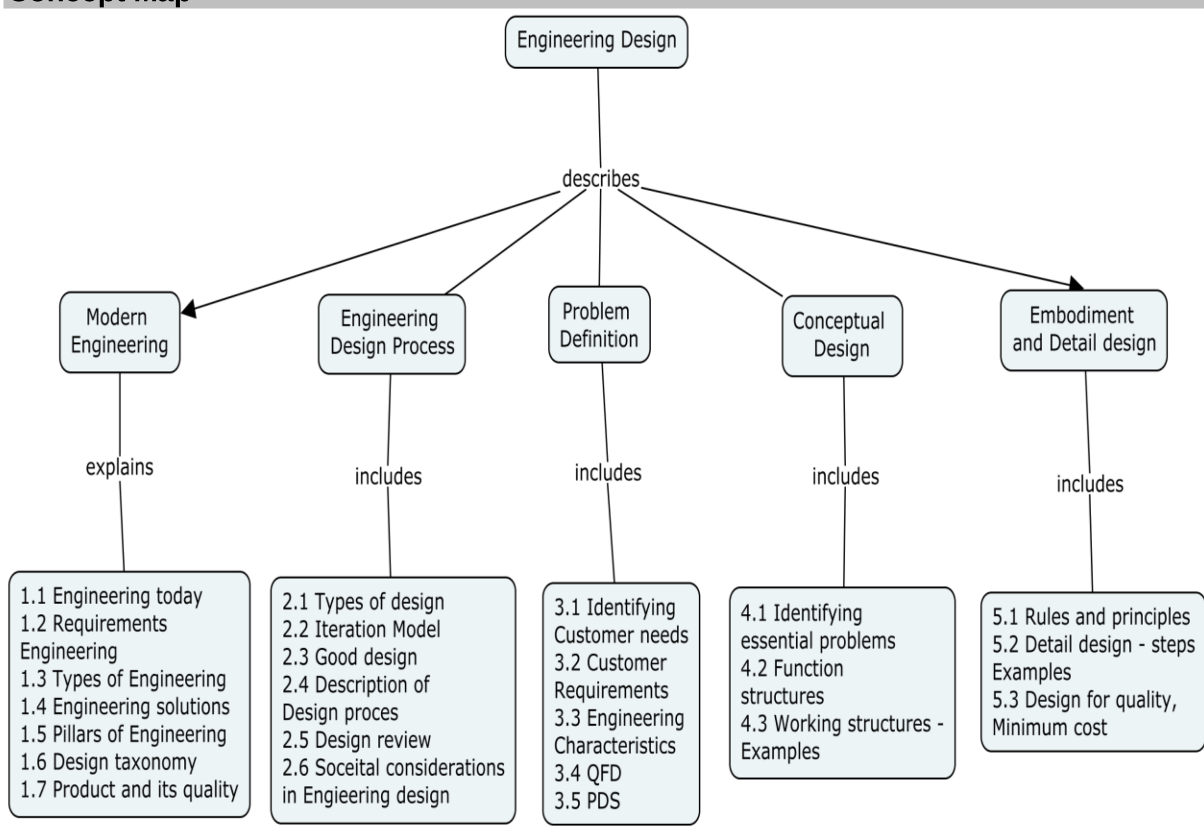
Course Outcome 3 (CO3)

1. Write product design specifications for any of the following product - Desktop Computer or Bicycle or Pencil or Computer Table or mobile.
2. Translate customer requirements into **Engineering characteristics** of any product like mobile or computer or bicycle.

Course Outcome 4 (CO4)

1. Prepare conceptual design document for any complex engineering problem related to societal engineering under specific domain.

Concept Map



Syllabus

Modern Engineering: Introduction, Engineering today, Requirements of engineering, Types of engineering, Engineering Solutions, Pillars of Engineering, Design Taxonomy, Product, Quality of product.
Engineering Design Process: Types of Designs, A Simplified Iteration Model, Considerations of a Good Design, Description of Design Process, Design Review, Societal Considerations in Engineering Design,
Problem Definition and Need Identification: Identifying Customer Needs, Customer Requirements, Establishing the Engineering Characteristics, Quality Function Deployment, product Design Specification

Conceptual Design: Steps, Abstracting to Identify the Essential Problems, Establishing Function Structures, Developing Working Structures and concepts. Examples
Embodiment and Detail Designs: Steps, Basic Rules and Principles of Embodiment Design, Detail Design, Design for Quality and minimum Cost. Examples

Reference Books

1. G.Pahl and W.Beitz (Translated by Ken Wallace et al.,) 'Engineering Design: A Systematic Approach, Second Edition, Springer, 2005.
2. George E. Dieter and Linda C. Schmidt, "Engineering Design", Fourth Edition, McGraw Hill Higher Education, 2009.
3. Power Point Presentation material by Prof.D.K.Subramanian in the Workshop on Engineering Design at TCE, Madurai.
4. Foundation Skills in Integrated Product Development, NASSCOM, Edition 2015

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Modern Engineering**	
1.1	Introduction - Engineering today	1
1.2	Requirements of engineering	
1.3	Types of engineering	1
1.4	Engineering Solutions	
1.5	Pillars of Engineering	
1.6	Design Taxonomy	1
1.7	Product and Quality of product	
2	**Engineering Design Process**	
2.1	Types of Designs	1
2.2	A Simplified Iteration Model	
2.3	Considerations of a Good Design	
2.4	Description of Design Process	1
2.5	Design Review	
2.6	Societal Considerations in Engineering Design	1
3	**Problem Definition and Need Identification**	
3.1	Identifying Customer Needs	1
3.2	Customer Requirements	
3.3	Establishing the Engineering Characteristics	
3.4	Quality Function Deployment	1
3.5	Product Design Specification	

Module No.	Topic	No. of Lectures
4	**Conceptual Design**	2
4.1	Steps, Abstracting to Identify the Essential Problems	
4.2	Establishing Function Structures	
4.3	Developing Working Structures and concepts - *Examples*	
5	**Embodiment and Detail Design**	2
5.1	Steps, Basic Rules and Principles of Embodiment Design	
5.2	Detail Design – *Examples*	
5.3	Design for Quality and minimum Cost	
	Total Lectures	12

Course Designers:

1. Dr.S.J.Thiruvengadam sjtece@tce.edu
2. Dr.S.Baskar sbeee@tce.edu

14CS350	DATA STRUCTURES AND ALGORITHMS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This course will cover data structures and the operations for manipulating them. Students will learn how to organize data so that data can be accessed and updated efficiently by a computer program.

Prerequisite

- 14CS270 : Problem Solving Using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Explain how arrays, stacks, queues, linked lists, trees, heaps, Graphs and Hash Tables are represented in the main memory and manipulated or used by different operations. (CO1)	Understand
Construct algorithms for performing operations on a data structure, with an understanding of the trade-off between the time and space complexity. (CO2)	Apply
Compare alternate implementations of an Abstract Data Type with respect to their performance. (CO3)	Analyze
Illustrate how arrays, stacks, queues, linked lists, trees, heaps, Graphs and Hash Tables are used in various applications. (CO4)	Understand
Analyze the computational efficiency of key searching, sorting and Hashing algorithms. (CO5)	Analyze
Evaluate the suitability of different data structures for solving computing problems. (CO6)	Evaluate

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M											
CO2	S	L	L									
CO3	S	S	M									
CO4	M	L										
CO5	S	S	M									
CO6	S	S	M									

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	

Remember	20	20	10	10
Understand	20	20	20	20
Apply	40	30	40	40
Analyse	20	20	20	20
Evaluate	0	10	10	10
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Outline the ways to implement three stacks in a single array.
2. Demonstrate the result of inserting the keys 10111101, 00000010, 10011011, 10111110, 01111111, 01010001, 10010110, 00001011, 11001111, 10011110, 11011011, 00101011, 01100001, 11110000, 01101111 into an initially empty extendible hash table with M = 4.
3. Outline a pseudo-code for creating a two dimensional linked list as shown in Fig. 1. The nodes in the first column contain only two pointers. The left pointer points to the next row and the right pointer points to the first data node in the row. Your solution should be capable of handling dynamic number of rows and columns.

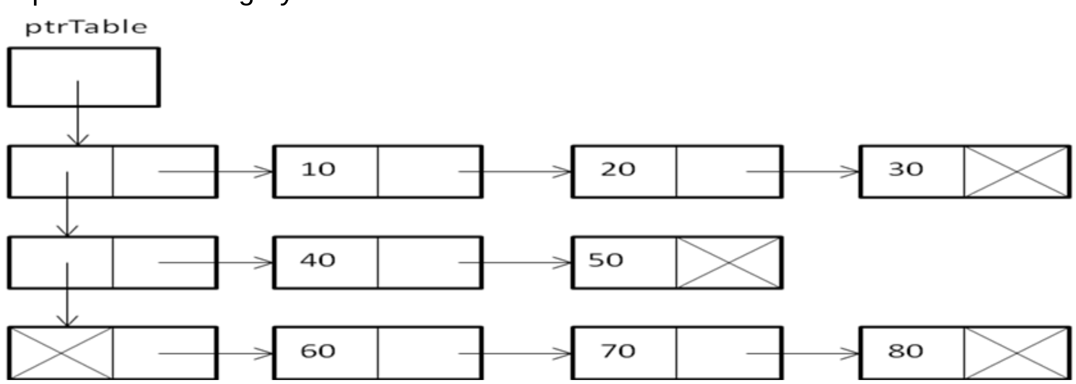


Fig. 1

4. Infer the performance of the following two splaying strategies in terms of the amortized cost incurred over a sequence of M insert, delete and search operations.

   - Strategy 1: Keep on rotating the key accessed with its parent by using a simple rotation strategy, till the key accessed reaches the root of the splay tree.
   - Strategy 2: Depending upon the structure of the sub-tree in which the key accessed is located, apply either zig-zig or zigzag rotation operation till the key accessed reaches the root.

Course Outcome 2 (CO2):

1. Given two sorted linked lists L1 and L2, construct a pseudo-code snippet to compute L1 ∩ L2 and to store the result in a third linked list L3. (Assume ascending ordering)
2. Construct a pseudo code for performing preorder traversal of a binary tree without using recursion.
3. Construct a pseudo code for inserting an element into a d-Heap and compute its time complexity.

4. Construct a pseudo-code to check if a linked list is circular. Assume that you are given with a pointer to an arbitrary node of the list and this pointer need not necessarily be the pointer to the head node of the list
5. Construct a recursive pseudo-code to perform insertion in an AVL tree. Demonstrate the different types of rotations involved to restore the balance of the tree.

Course Outcome 3 (CO3)

1. Compare and contrast the following data structures with respect to their capability to act as a priority queue.
   i) Binary heap
   ii) Leftist heap
   iii) Binomial queue

2. A deque is a data structure consisting of a list of items, on which the following operations are possible:
   push(X, D): Insert item X on the front end of deque D.
   pop(D): Remove the front item from deque D and return it.
   Inject(X,D): Insert item X on the rear end of deque D.
   Eject(D): Remove the rear item from deque D and return it.
   Analyze the requirements and propose an implementation to support all these operations in O(1) time per operation.

3. Compare and contrast the following data structures with respect to their suitability in implementing the Stack ADT.
   i) Array
   ii) Singly Linked List

Course Outcome 4 (CO4)

1. Assume that a backtracking algorithm which uses a stack has been designed to find a solution to the 4-Queen's problem on a 4x4 chess board. The algorithm has already completed an unknown number of iterations say 'k'. After these 'k' iterations, the chess board state and the state of the stack are as shown in Fig. 2 and Fig. 3 respectively. Infer the sequence of chess board configurations and the sequence of stack states for each of the iterations from 1 to k.

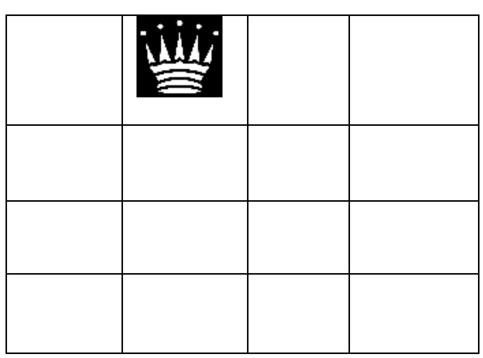


Fig. 2

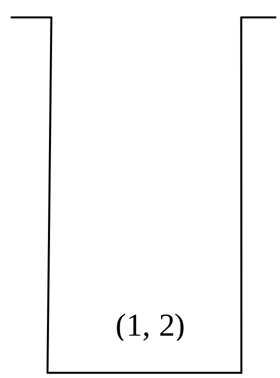


Fig. 3

2. Outline a pseudo-code to multiply two polynomials. Assume that the polynomials are stored in the form of a singly linked list in memory. Make sure that the output polynomial is sorted by exponent and has at most one term of any power.

3. Assume that a goal-seeking application is being written whose job is to find a path from the source city '1' to the destination city '12' in the map [Fig 1] given below:

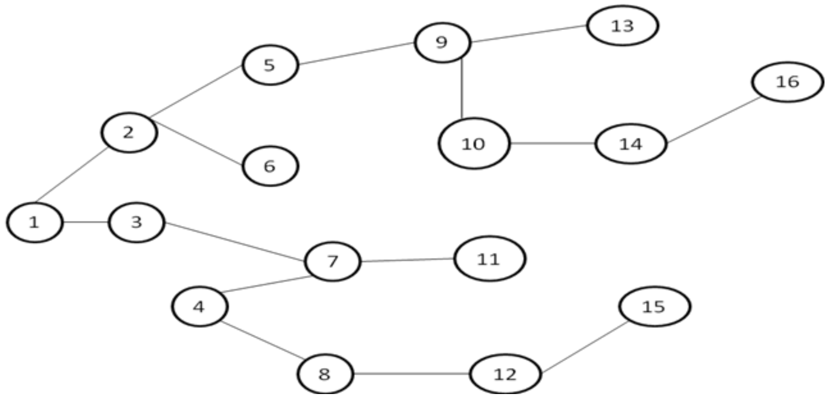


Fig. 4

In Fig. 4, nodes represent cities and lines represent roads between the cities. Assume that this application has a stack to implement the goal seeking function using backtracking. Illustrate the sequence of steps involved in the backtracking algorithm to trace a path from the source city to the destination city. Infer the various states in which the stack would have been during an execution of this algorithm.

Course Outcome 5 (CO5)

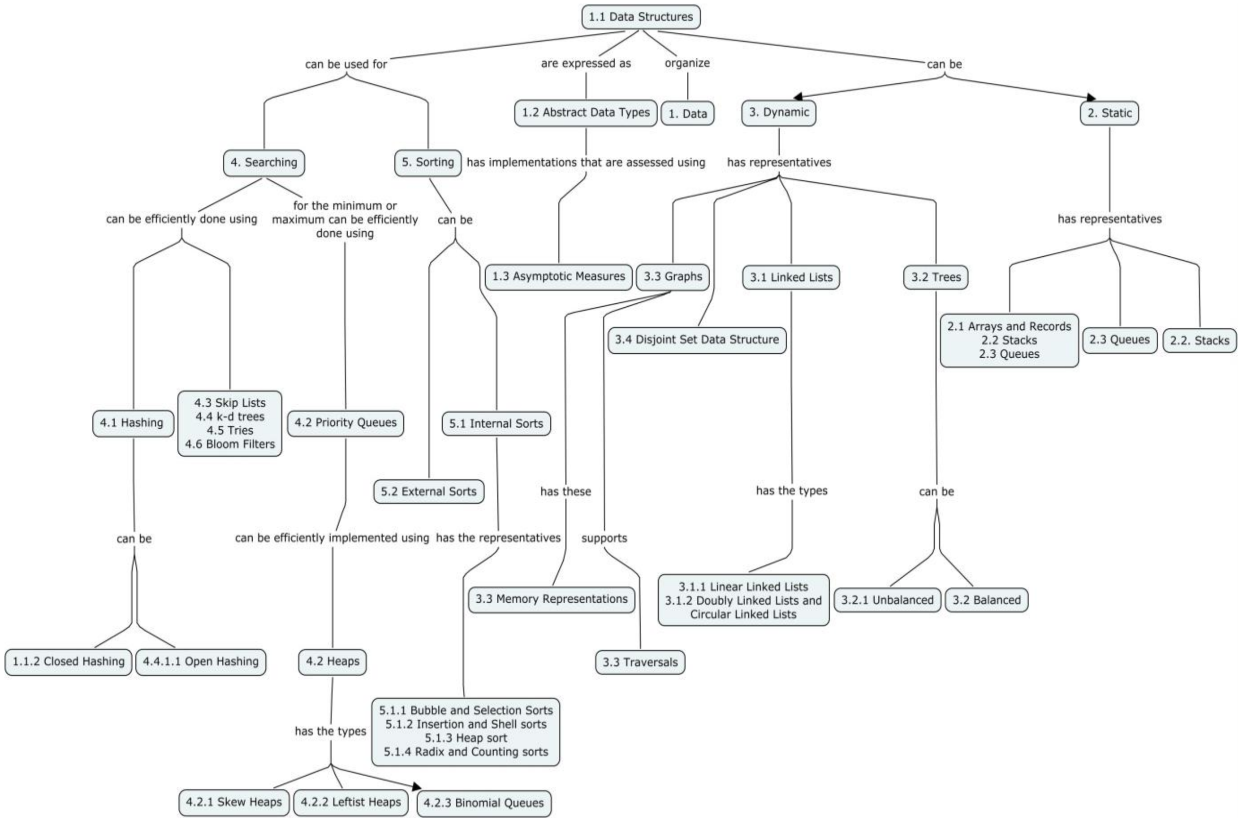
1. Compare and contrast balanced merge and poly-phase external merge sorts.
2. Examine the running time of insertion sort for i) Sorted input, ii) Reverse ordered input.
3. Evaluate the best-case, average-case and worst-case time complexity of the quick sort algorithm with the help of a pseudo-code snippet and recurrence equations.
4. Defend through a precise mathematical argument that the height of an AVL tree is $O(\log_2 n)$, where n is the number of nodes in the AVL tree.
5. Examine the pros and cons of the following hash function:

```
int hash(const char *key, int tableSize) {
        return (key[0] + key[1] * 27 + key[2] * 272) % tableSize;      }
```

Course Outcome 6 (CO6)

1. Assume that a spell-checker program needs to be written so that it prints out all words not in some online dictionary. Suppose the dictionary contains 30,000 words and the file is large, so that the algorithm can make only one pass through the input file. Analyze which data structure will be suitable for implementing this spell-checker application in terms of time and space complexities.
2. Recommend a suitable data structure to evaluate prefix expressions and depict the sequence of steps to be followed in evaluating a prefix expression using your recommendation.
3. Recommend an efficient data structure to add two polynomials (having M and N terms). You must make sure that the output polynomial is sorted by exponent and has at most one term of any power. Assume that the polynomials are of the form $a_1x^m + a_2x^{m-1} + \ldots + a_m$ and $b_1x^n + b_2x^{n-1} + \ldots + b_n$

Concept Map



Syllabus

Data: Data Structure, Abstract Data Types, Asymptotic Measures **Static Data Structures:** Arrays and Records**,** Stacks, Queues **Dynamic Data Structures:** Linked Lists: Linear Linked Lists, Doubly Linked Lists and Circular Linked Lists, Trees: Unbalanced and Balanced Trees, Graphs - Memory Representations and Traversals, Disjoint set data structure. **Searching:** Hash Tables: Open Hashing and Closed Hashing; Priority Queues - Heaps: Skew Heaps, Leftist Heaps and Binomial Queues; Skip Lists, Tries, k-d trees, Bloom Filters **Sorting:** Internal Sorts: Bubble and Selection sorts, Insertion and Shell sorts, Heap sort, Radix and Counting sorts, External Sorts.

Text Book

1. Richard F. Gilberg , Behrouz A. Forouzan: Data Structures: A Pseudocode Approach With C, 2nd Edition, Thomson Learning, 2003

References

1. Mark Allen Weiss: Data Structures and Algorithms in C, Addison-Wesley, 1997
2. Lecture notes hosted on TCE-Moodle for course modules 4.3 to 4.6

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Data**	
1.1	Data Structure	1

Module No.	Topic	No. of Lectures
1.2	Abstract Data Types	1
1.3	Asymptotic Measures	1
2	**Static Data Structures**	
2.1	Arrays and Records	1
2.2	Stacks	1
2.3	Queues	1
3	**Dynamic Data Structures**	
3.1	Linked Lists	1
3.1.1	Linear Linked Lists	1
3.1.2	Doubly Linked Lists and Circular Linked Lists	1
3.2	Trees	1
3.2.1	Unbalanced Trees	2
3.2.2	Balanced Trees	2
3.3	Graphs - Memory Representations and Traversals	2
3.4	Disjoint set data structure	1
4	**Searching**	
4.1	Hashing	1
4.1.1	Open Hashing	2
4.1.2	Closed Hashing	2
4.2	Priority Queues - Heaps	1
4.2.1	Skew Heaps	1
4.2.2	Leftist Heaps	1
4.2.3	Binomial Queues	1
4.3	Skip Lists	1
4.4	Tries	1
4.5	k-d trees	1
4.6	Bloom Filters	1
5	**Sorting**	
5.1	Internal Sorts	
5.1.1	Bubble and Selection sorts	1
5.1.2	Insertion and Shell sorts	1
5.1.3	Heap sort	1
5.1.4	Radix and Counting sorts	1
5.2	External Sorts	2
	Total	36

Course Designers:

1. Mr. S. Karthick skcse@tce.edu
2. Dr. M. K. Kavitha Devi mkkdit@tce.edu

		Category	L	T	P	Credit
14CS370	**OBJECT ORIENTED PROGRAMMING**	PC	2	0	1	3

Preamble

This syllabus is intended for the Computer science students and enables them to learn Object Oriented Programming and the design of computer solutions in a precise manner. The syllabus emphasizes on OOP concepts, Functions, Polymorphism, Inheritance and I/O. The intention is to provide sufficient depth in these topics to enable candidates to apply Object Oriented Programming approach to programming. Besides the written papers, lab-based examinations are included as part of the assessment requirements for the study. The lab-based examinations will test the candidate's ability to develop computer-programming solutions for a series of programming tasks of varying complexity. The modules in the syllabus reflect solving general problems via programming solution. Thus, modules collectively focus on programming concepts, strategies and techniques; and the application of these toward the development of programming solutions.

Prerequisite

- 14CS270 : Problem Solving using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Construct object-oriented programs for a given scenario using the concepts of abstraction, encapsulation, message-passing and modularity. (CO1)	Apply
Develop object-oriented programs for a given application using the concepts of compile-time and run-time polymorphism. (CO2)	Apply
Construct object-oriented programs for a given application by demonstrating the inter-relationship between classes using inheritance and aggregation. (CO3)	Apply
Develop object-oriented applications that can handle exceptions. (CO4)	Apply
Construct object-oriented applications for a given scenario to persist data using files and object-serialization. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	S	S					M	M	L	L	L
CO2	S	S	S					M	M	L	L	L
CO3	S	S	S					M	M	L	L	L
CO4	S	S	S					M	M	L	L	L
CO5	S	S	S					M	M	L	L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3(Practical)	Theory
Remember	30	20	-	20
Understand	30	20	-	20
Apply	40	60	100	60
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define a method signature. (Remember)
2. List the difference between inline and non-inline functions. (Remember)
3. List the strategy used by C++ and Java in passing arguments to a Function / method (Remember)
4. Write about the benefits of Object oriented Programming (Understand)
5. Write a Java program to create 2 two-dimensional arrays which hold numbers. Write a method which takes the arrays as arguments to perform matrix multiplication. (Apply)
6. Assume a class in C++ named Car that keeps track of price of cars. It has member variables carName, price and taxRate. Write a member function that computes the total price(including tax) of the Car object with the values passed as arguments, but which also includes 12.5% as a reasonable default value for taxRate. Create 2 Car objects and display their total price. (Apply)

Course Outcome 2 (CO2):

1. Recollect the term compile time polymorphism. (Remember)
2. Define overriding. (Remember)
3. Discuss in detail about run time polymorphism. (Understand)
4. Consider a class called **Dealer** and a class called **PartSupplier**. Multiply inherit them into **DealerPartSupplier**. Now define **LocaldealerPartSuppler** and **OutStationdealerPartSupplier** inheriting from DealerPartSupplier. Use virtual functions in C++ to show the number of parts supplied by LocaldealerPartSupplier and OutStationdealerPartSupplier. (Apply)

5. Write a C++ program to create 2 overloaded functions named findAt to find the digit / character at the position specified. One function takes a number and position as arguments. Another function takes a string and position as arguments. (Apply)

Course Outcome 3 (CO3):

1. State the difference between aggregation and composition. (Remember)
2. Explain about Inheritance hierarchies with an example program. (Understand)
3. Discuss about public and protected derivations with an example program. (Understand)
4. Write a class implementation in Java to demonstrate the aggregate relationship between a **Department** and a **Course.** The Department offers several courses and each course has an associated credit. Design methods in the Department class to calculate the number of courses offered and the total number of credits. Design methods in the Course class to set and get its credit. (Apply)
5. Consider the scenario in which a class named **EBCustomer** that inherits into **Domestic** and **Commercial**. Define an interface named calculateTariff that is used both by the Domestic EB customer and the Commercial EB customer in calculating the Nett current consumption charges. Create a Domestic EB customer object and a Commercial EB customer object and print the Nett current consumption charges of the customers. (Apply)

Course Outcome 4 (CO4):

1. Define a checked and an unchecked exception. (Remember)
2. Explain the exception handling mechanism in C++. (Understand)
3. Write a class in C++ called Date which stores a date in three integers: month, day and year. There should be member functions to print the date in the following formats (dd/mm/yy) and (dd/month-string/yy). The class should implement the following exception classes.
   - InvalidDay throw when an invalid day(<1 or >31) is passed to the class
   - InvalidMonth throw when an invalid month(<1 or >12) is passed to the class

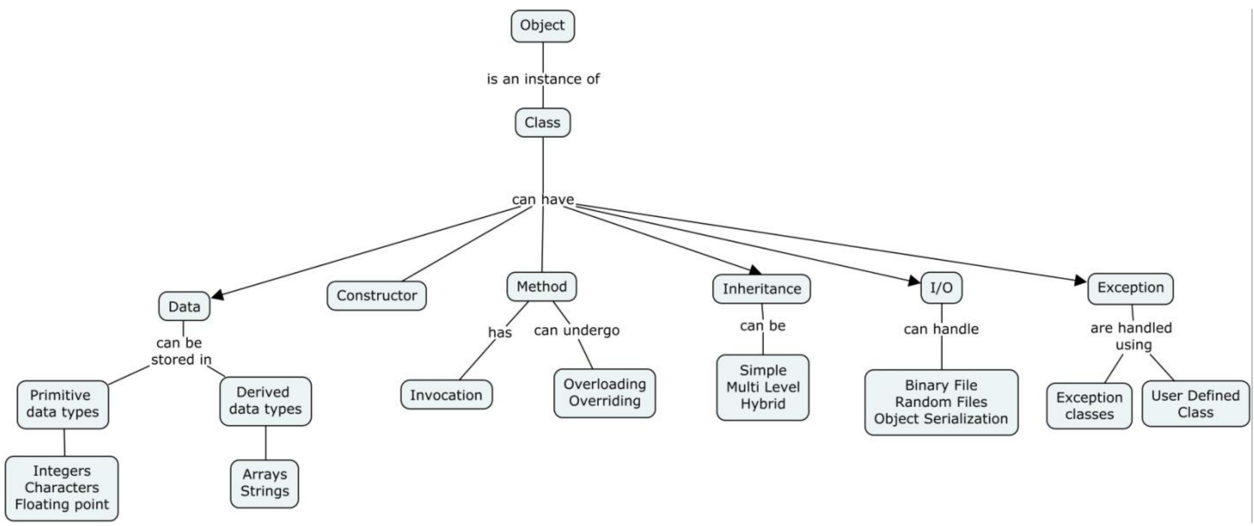
   (Apply)

Course Outcome 5 (CO5):

1. List any 4 methods of Data Input stream class. (Remember)
2. Discuss the advantages of Random AccessFiles using a sample program. (Understand)

3. Write a program that uses a structure to store the following student information in a file. The program should have a menu that allows the user to perform the following tasks i) Add new records to the file ii) Display any record in the file. (Apply)
4. Consider a class called **UsedCar** which uses an interface called secsalesItem. The UsedCar class has the following fields: vehicleNumber, model, year, kmTravelled, price and the following methods: getVehicleNumber, getModel and getRetailPrice. The interface secsalesItem has only one method called getRetailPrice which calculates price based on the year of manufacture and kilometers travelled. Create 5 usedCar objects and serialize them into a file. (Apply)

Concept Map



Syllabus

Basics of Object oriented Programming Object oriented programming and its benefits - Object oriented programming concepts: Encapsulation, Information hiding and Abstraction – Object oriented programming concepts: Generalization/Specialization and Polymorphism – Object Models - **Methods-**Passing Arguments to a Method - Returning a Value from a Method - Method overloading **Classes and Objects-**Object oriented design: finding the Classes and their Responsibilities - Instance fields and Methods-Constructors and overloading - Passing Arrays As Arguments to Methods - Returning Arrays from Methods - Passing Objects to Methods- Returning Objects from Methods - Arrays of Objects **Class collaborations and Polymorphism** Object oriented Design: Class Collaborations – Aggregation - Chains of Inheritance – Interfaces - Overriding Super class methods - Abstract Classes and Abstract

Methods **I/O Handling and Exception Handling -** Binary files - Random-Access files- Object serialization – Exception handling

Text Books

1. Tony Gaddis, Judy Walters and Godfrey Muganda: Starting out with Object Oriented Programming in C++, 3rd Alternate Edition, Dreamtech Press, 2006.
2. Tony Gaddis, Starting Out with Java: From Control Structures through Objects, 4/E, Addison-Wesley, 2009.
3. Grady Booch, Robert Maksimchuk, Michael Engel, Bobbi Young, Jim Conallen, Kelli Houston: Object Oriented Analysis and Design with Applications, Third Edition, May 2007.

Reference Book

1. H.M. Deitel and P.J. Deitel, C How to program Introducing C++ and Java, Fourth Edition, Pearson Prentice Hall, 2005.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Basics of Object oriented Programming (3)**	
1.1	Object oriented programming and its benefits	1
1.2	Object oriented programming concepts: Encapsulation, Information hiding and Abstraction	1
1.3	Object oriented programming concepts: Generalization/Specialization and Polymorphism, Object Models	2
2	**Methods(2)**	
2.1	Passing Arguments to a Method and Returning a Value from a Method	1
2.2	Method overloading	1
3	**Classes and Objects(8)**	
3.1	Object oriented design: finding the Classes and their Responsibilities	1
3.2	Instance fields and Methods, Constructors and overloading	1
3.3	Passing Arrays As Arguments to Methods	1
3.4	Returning Arrays from Methods	1
3.5	Passing Objects to Methods	1
3.6	Returning Objects from Methods	1
3.7	Arrays of Objects	1
4	**Class collaborations and Polymorphism(6)**	
4.1	Object oriented Design: Class Collaborations	1
4.2	Aggregation	1
4.3	Chains of Inheritance and Interfaces	2
4.4	overriding super class methods	1
4.5	Abstract Classes and Abstract Methods	1

Module No.	Topic	No. of Lectures
5	**I/O Handling and Exception Handling (5)**	
5.1	Binary files	1
5.2	Random-Access files	1
5.3	Object serialization	1
5.4	Exception handling	2

Module No.	Topic	No. of Lectures
1	Develop Object Oriented Program for passing arguments to a method and returning value from a method	2
2	Construct Object Oriented Program for method overloading and constructor overloading	2
3	Develop Object Oriented Program for passing arrays and objects as arguments to method and returning arrays and objects from methods	4
4	Demonstrate aggregation using object oriented program	2
5	Develop Object Oriented Program to demonstrate inheritance and overriding super class methods	4
6	Develop Object Oriented Program to demonstrate abstract base classes abstract methods.	2
7	Construct Object Oriented Program to demonstrate File handling and Object Serialization	4
8	Construct Object Oriented Program to demonstrate exception handling	4

Course Designers

1. Mrs.J.Jane Rubel Angelina janerubel@tce.edu
2. Mr.S.Prasanna sprcse@tce.edu

		Category	L	T	P	Credit
14CS380	**DATA STRUCTURES LAB**	PC	0	0	1	1

Preamble

With a dynamic learn-by-doing focus, this laboratory course encourages students to explore data structures by implementing them, a process through which students discover how data structures work and how they can be applied. This course challenges students to exercise their creativity in both programming and analysis.

Prerequisite

- 14CS270 : Problem Solving Using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Construct implementations for Abstract Data Types (ADT) using appropriate Data Structures (CO1)	Apply
Implement solutions for engineering problems by using appropriate data structures. (CO2)	Apply
Assess the suitability of a data structure to solve a problem, based on the time and space complexities of different operations on the data structure (CO3)	Evaluate
Demonstrate problem-solving and programming skills while solving problems (CO4)	Apply
Implement algorithms which use sorting, searching and/or selection as sub-procedures.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L		L				M	L		
CO2	S	M	L		L				M	L		
CO3	S	M	L		L				M	L		
CO4	S	M	L		L				M	L		M
CO5	S	M	L		L				M	L		

S- Strong; M-Medium; L-Low

Course Level Assessment Questions

Course Outcome 1 (CO1, CO4):

1. A self-adjusting list is like a regular list, except that all the insertions are performed at the front, and when an element is accessed by a find it is moved to the front of the list

without changing the relative order of the other items. Construct a linked list implementation of a self-adjusting list.

2. A deque is a data structure consisting of a list of items, on which the following operations are possible:
   push(X, D): Insert item X on the front end of deque D.
   pop(D): Remove the front item from deque D and return it.
   Inject(X,D): Insert item X on the rear end of deque D.
   Eject(D): Remove the rear item from deque D and return it.
   Analyze the requirements and propose an implementation to support all these operations in O(1) time per operation.

Course Outcome 2 (CO2, CO4):

1. Given two sorted linked lists L1 and L2, construct an implementation to compute L1 ∩ L2 and to store the result in a third linked list L3.

2. Assume that a backtracking algorithm has to be designed to solve the N-Queens problem using a stack. The algorithm has completed an unknown number of passes in a 4 x 4 chess board at which point, the state of the chessboard is as shown in Fig. 1. Infer the sequence of stack states and chess board configurations from the first pass till the current pass by implementing the backtracking process.

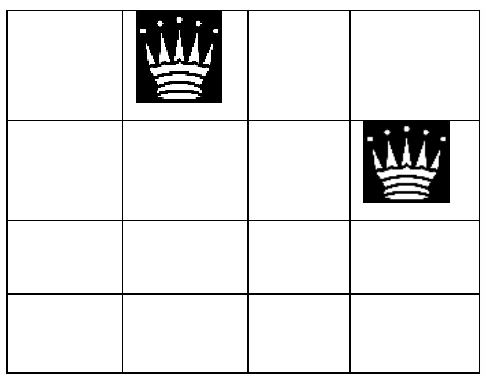


Fig. 1

Course Outcome 3 (CO3, CO4)

1. Recommend a way to implement an ordered singly linked list so that search operation can be supported in less than O(n) worst case time (preferably in O(log N)). 'n' is the number of items in the list.

2. With help of a call-tree/call-stack evaluate the following strategies (S1 and S2) in terms of their space requirements and computational (time) complexity.
   S1: A recursive algorithm to determine the minimum element in a Binary Search Tree.
   S2: An iterative (non-recursive) algorithm to determine the minimum element in a Binary Search Tree.

3. Implement a Suffix Trie and employ it to find the longest common substring between two strings.

List of Experiments

1. Stack implementation and Two-way stack implementation using arrays
2. Evaluation of a postfix expression using a Stack
3. Implementation of a Queue, Circular Queue using an Array
4. Round-robin Scheduling using a circular queue
5. Implementation of Singly, Doubly and Circular Linked Lists.
6. Polynomial Arithmetic using Linked Lists
7. Binary Search Tree operations and Traversals
8. Implementation of Hash Tables and Bloom filters
9. Implementation of Suffix Tries for fast full-text searches
10. Binary Heaps and their applications
11. Internal Sorts (Bubble, Selection, Shell, Heap, Quick, Merge, Counting, Radix)
12. Implementation of an application by selecting or constructing a data structure to suit the requirements of the application and measuring the performance of the application in terms of the running time and space consumed.

Course Designers:

1.	Mr. S. Karthick	skcse@tce.edu
2.	Dr. M. K. Kavitha Devi	mkkdit@tce.edu

14CS390	ASSEMBLY LANGUAGE PROGRAMMING LABORATORY	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

This course is deigned to enable the students to develop assembly language programs for simple arithmetic operations, code conversions, carry out interfacing and execute BIOS and DOS interrupt service routines. These experiments are aimed at imparting a practical exposure to the students to prepare them for low level programming which directly interacts with the underlying hardware.

Prerequisite

- 14CS230 : Digital Circuits
- 14CS280 : Digital Circuits Laboratory
- 14CS240 : Computer Organization and Microprocessors

Course Outcomes

On the successful completion of the course, students will be able to

Develop assembly language programs for binary arithmetic and code conversions (CO1)	Apply
Develop assembly language programs for accessing the parallel ports of a PC for interfacing and waveform generation (CO2)	Apply
Develop assembly language programs for executing DOS and BIOS interrupts for specific purposes (CO3)	Apply
Develop assembly language programs with an understanding of the underlying microprocessor architecture, instruction set and addressing modes. (CO4)	Apply
Implement assembly language programs using an assembler or a microprocessor kit with an understanding of its features like pseudo-instructions and debuggers (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	L	S		M					L		
CO2	S	L	S		M					L		
CO3	S	L	S		M					L		
CO4	S	L	S		M					L		
CO5	S	L	S		S					L		

S- Strong; M-Medium; L-Low

List of Experiments

1. Addition of an array of 16-bit numbers with carry.
2. Addition and Multiplication of 32-bit numbers.

3. Addition and multiplication of unpacked BCD numbers.
4. Multiplication of two 8-bit packed BCD numbers.
5. 16-bit binary to gray code conversion.
6. 16-bit binary to excess 3 code
7. Sorting of 16-bit numbers
8. Addition of two 2*2 matrices
9. ADC and DAC interface with PC
10. Stepper interface with PC
11. Use of BIOS and DOS interrupts.
12. Generation of waveforms

Course Designers:

1. Mr. C.Sridharan cscse@tce.edu
2. Mr. R.Chellamani rcmcse@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

FOURTH SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY
(For the candidates admitted from 2016-17 onwards)

FOURTH SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS410	Discrete Mathematics and Combinatorics	BS	2	2	-	3
14CS421	System Software and Operating Systems	PC	3	-	-	3
14CS430	Design and Analysis of Algorithms	PC	3	-	-	3
14CS440	Database Management Systems	PC	3	-	-	3
14CS450	Communication Engineering	PC	3	-	-	3
THEORY CUM PRACTICAL						
14CS470	Professional Communication	HSS	1	-	2	2
PRACTICAL						
14CS480	System Software and Operating Systems Lab	PC	-	-	2	1
15CS490	Algorithms Lab	PC	-	-	2	1
SPECIAL COURSES						
14CS4C2	Capstone Course – I	PC	-	-	2*	2
	Total		**15**	**2**	**8**	**21**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core
L : Lecture
T : Tutorial
P : Practical

Note:
1 Hour Lecture is equivalent to 1 credit
2 Hours Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

* - 2 hours/ week is allotted for off-class practical work

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

FOURTH SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS410	Discrete Mathematics and Combinatorics	3	50	50	100	25	50
2	14CS421	System Software and Operating Systems	3	50	50	100	25	50
3	14CS430	Design and Analysis of Algorithms	3	50	50	100	25	50
4	14CS440	Database Management Systems	3	50	50	100	25	50
5	14CS450	Communication Engineering	3	50	50	100	25	50
THEORY CUM PRACTICAL								
7	14CS470	Professional Communication	3	50	50	100	25	50
PRACTICAL								
8	14CS480	System Software and Operating Systems Lab	3	50	50	100	25	50
9	15CS490	Algorithms Lab	3	50	50	100	25	50
10	14CS4C2	Capstone Course - I	-	100	-	100	-	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

14CS410	DISCRETE MATHEMATICS AND COMBINATORICS	Category	L	T	P	Credit
		BS	2	1	0	3

Preamble

A course in discrete mathematics teaches students how to work with discrete structures, which are the abstract mathematical structures used to represent discrete objects and relationships between these objects. These discrete structures include logic, predicate calculus and sets. An important problem–solving skill is the ability to count or enumerate objects. The discussion of enumeration in this course begins with basic techniques of counting. The general counting methods involve permutations and combinations. These methods are very useful in constructing computer programs and in mastering many theoretical topics of computer science. Recurrence relations are one of the simplest ways to solve counting problems. The methods for solving recurrence relations appeared originally in the development of the theory of difference equations, cousins of differential equations.

Prerequisite

Basic Set Language

Course Outcomes

On the successful completion of the course, students will be able to

Rephrase real world statements as logical propositions and demonstrate whether the proposition is satisfiable, tautology or a contradiction. (CO1)	Apply
Infer whether a logical argument is valid from the given set of premises by applying the inference rules of predicate calculus. (CO2)	Apply
Show mathematical reasoning and arrive at conclusions about sets and relations (CO3)	Apply
Use the concepts of sets and relations as tools to reduce complexity. (CO4)	Apply
Construct the number of arrangements and selections using the principles of counting. (CO5)	Apply
Construct the recurrence relation for a given engineering problem and solve the recurrence equation. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L									
CO2	S	M	L									
CO3	S	M	L									
CO4	S	M	L									
CO5	S	M	L									
CO6	S	M	L									

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	10	10	10	0

Understand	30	30	30	30
Apply	60	60	60	70
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define a biconditional statement and draw its truth table.
2. Estimate the PCNF and PDNF of the formula given by $(7P \rightarrow R) \wedge (Q \rightarrow P)$.
3. Show that following implication $[P \rightarrow (Q \rightarrow R)] \Rightarrow [(P \rightarrow Q) \rightarrow (P \rightarrow R)]$.

Course Outcome 2 (CO2):

1. Show that $S \vee R$ is tautologically implied by $(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow S)$ using automatic theorem proving.
2. Show that $(x)(P(x) \vee Q(x)) \Rightarrow (x)P(x) \vee (\exists(x))Q(x)$,Using indirect method.
3. Test the validity of the following argument

If Ram is clever, the sam is well-behaved. If Joe is good, then sam is bad and prem is not well- behaved. If Lal is educated, then Joe is good or Ram is clever. Hence if Lal is educated and prem is not well-behaved then sam is bad.

Course Outcome 3 (CO3):

1. Show that $A - B = A \cap B'$.
2. Let R denote a relation on the set of ordered pairs of integers such that $< x, y > R < u, v >$ iff xv=yu. Show that R is an equivalence relation
3. . Let f(x)=x+2, g(x)=x-2,h(x)=3x, for $x \in R$, where R is the set of real numbers. Find $f \circ g, g \circ f, f \circ f, g \circ g$ and $f \circ g \circ h.$ And also write the corresponding matrix.

Course Outcome 4 (CO4):

1. If A={c,d}, B={1,2}, C={2,3} Find A X (BUC).
2. Let the compatibility relation on a set {1,2,3,4,5} given by the matrix . Then find the maximal compatible set

$$
\begin{array}{c|cccc}
2 & & & & \\
3 & 0 & & & \\
4 & 1 & 1 & & \\
5 & 1 & 0 & 1 & \\
6 & 0 & 1 & 0 & 1 \\
\hline
 & 1 & 2 & 3 & 4
\end{array}
$$

3. Given S = {1,2,3,-----,10} and a relation R on S where R = {(x,y) / x+y = 10} what are the properties of the relation R?

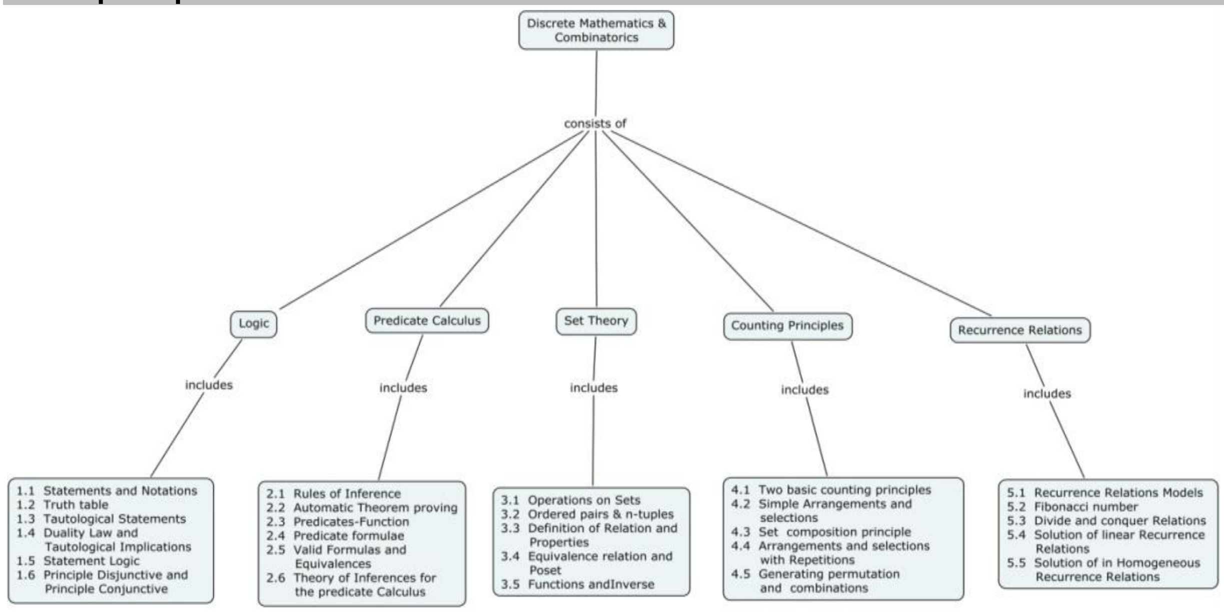
Course Outcome 5 (CO5):

1. State two basic counting principles.
2. A test contains 100 true/false questions. Compute how many different ways can a student answer the questions on the test, if answers may be left blank?
3. There are 12 signs of the zodiac. Determine how many people are needed to guarantee that at least six of these people have the same sign?

Course Outcome 6 (CO6):

1. Find the recurrence relation for the sequence $\{a_n\}$ given by $a_n = A.2^n+B.(-3)^n$.
2. A person climbs a staircase by climbing either(i) two steps in a single stride or (ii) only one step in a single stride. Find a recurrence relation for the number of ways of climbing n stairs.
3. An Italian mathematician, proposed the following problem around the year 1202: If
(i)There are two new born rabbits, one male and one female, at the beginning;
(ii) Each mixed pair gives birth to a mixed pair at the end of two months and in each month afterwards.
(iii) No rabbit dies. Find the number of pairs of rabbits at the beginning of the nth month.
4. Find the general solution of the recurrence relation if 1,2,3,4,4,4,1+I,1-I are the roots of the corresponding characteristic equation.
5. Solve the recurrence relation $a_n-a_{n-1}-12a_{n-2}=0$, $a_0=0$, $a_1=1$.
6. Find the total solution of the recurrence relation $a_n-5a_{n-1}+6a_{n-2}=2^n+3n$, $a_0=1$, $a_1=6$.

Concept Map



Syllabus

Logic: Introduction – Statements and Notations , Negation – Conjunction – Disjunction – Truth table , Conditional – Biconditional – Tautological Statements – Equivalence of Formulas, Duality Law – Tautological Implications-Functionally Complete set of Connectives –Other Connectives, Two State Devices & Statement Logic, Disjunctive – Conjunctive – Principle Disjunctive – Principle Conjunctive. **Predicate calculus:** Checking the validity using the truth table, Rules of Inference –Consistency of premises and Indirect Method, Automatic Theorem proving, Predicates-Function, Variables and Quantifiers, Valid Formulae – Equivalences, Theory of Inferences for the predicate Calculus. **Set Theory:** Basic Definitions – Operations on Sets – Identities, Ordered pairs & n-tuples –Cartesian Products, Definition of Relation –Binary Relation – Properties-Matrix – Graph, Equivalence relation – Compatibility Relation-Composition of relation - Poset, Functions –Composition-Inverse. **Counting principles:** Two basic counting principles , Simple Arrangements and selections, Set composition principle, Arrangements and selections with Repetitions , Generating permutation and combinations. **Recurrence Relations:** Recurrence Relations Models, Fibonacci number, Divide and conquer Relations, Solution of linear Recurrence Relations, Solution of in Homogeneous Recurrence Relations.

Text Books

1. T.P.Tremblay and R.Manohar , "Discrete Mathematical Structures with application to Computer Science",Tata McGraw Hill, 2002.
2. Alan Thucker, "Applied Combinatorics" ,John Wiley & Sons, Incorporated, 2003.

Reference Books

1. Dr.M.K.Venkataraman., Dr.N.Sridharan and N.Chandrasekaran, " Discrete Mathematics ", National Publishing Company, India, 2004.
2. Lin,C.L. " Introduction to Combinatorial Mathematics", McGraw Hill Book Company, New York, 1968.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Logic**	
1.1	Introduction – Statements and Notations	1
1.2	Negation – Conjuction – Disjunction – Truth table	1
	Tutorial	1
1.3	Conditional – Biconditional – Tautological Statements – Equivalence of Formulas	1
	Tutorial	2
1.4	Duality Law – Tautological Implications-Functionally Complete set of Connectives –Other Connectives	1
1.5	Two State Devices & Statement Logic	1
	Tutorial	1
1.6	Disjunctive – Conjunctive – Principle Disjunctive	1

Module No.	Topic	No. of Lectures
	– Principle Conjunctive	
	Tutorial	2
2	**Predicate calculus**	
2.1	Checking the validity using the truth table, Rules of Inference –Consistency of premises and Indirect Method	1
2.2	Automatic Theorem proving	1
	Tutorial	1
2.3	Predicates-Function, Variables and Quantifiers	1
	Tutorial	1
2.4	Predicate formulae – Free and Bound Variables, Valid Formulas – Equivalences	1
	Tutorial	1
2.5	Theory of Inferences for the predicate Calculus	1
	Tutorial	2
3	**Set Theory**	
3.1	Basic Definitions – Operations on Sets – Identities, Ordered pairs & n-tuples –Cartesian Products	1
	Tutorial	2
3.2	Definition of Relation –Binary Relation – Properties-Matrix - Graph	1
	Tutorial	1
3.3	Equivalence relation – Compatibility Relation- Composition of relation- Poset	1
3.4	Functions –Composition-Inverse	1
	Tutorial	2
4	**Counting principles**	
4.1 4.2	Two basic counting principles Simple Arrangements and selections	1
	Tutorial	2
4.3	Set composition principle	1
	Tutorial	1
4.4	Arrangements and selections with Repetitions	1
4.5	Generating permutation and combinations	1
	Tutorial	1
5	**Recurrence Relations**	
5.1	Recurrence Relations Models	1
5.2	Fibonacci number	1
	Tutorial	1
5.3	Divide and conquer Relations	1
5.4	Solution of linear Recurrence Relations	1
	Tutorial	1
5.5	Solution of in Homogeneous Recurrence Relations	1
	Tutorial	2
	Total	**48**

Course Designer

1. Mr. B. Vigneshwaran vigneshwaran@tce.edu

14CS421	SYSTEM SOFTWARE AND OPERATING SYSTEMS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

To provide a basic knowledge in the concepts of system software and to understand the working principle of assemblers, functions of loaders, linkers and macro processors. Operating systems is an essential part of any computer-science education. Although this field is undergoing rapid change, as computers are now prevalent in virtually every application, the fundamental concepts remain fairly clear. It provides a clear description of the *concepts* that underlie operating systems. The fundamental concepts and algorithms are based on those used in existing commercial operating systems.

Prerequisite

14CS240 : Computer Organization and Microprocessor

Course Outcomes

On the successful completion of the course, students will be able to

Describe the functions and types of assemblers, linkers and loaders. (CO1)	Understand
Develop single-pass and multi-pass macro processors. (CO2)	Understand
Develop programs using system-calls related to process, memory and file management. (CO3)	Apply
Construct solutions for problems related to process scheduling, deadlocks and synchronization in a multi-programmed operating system. (CO4)	Apply
Develop appropriate solutions for memory management considering challenges due to multi-programming and virtual memory. (CO5)	Apply
Construct solutions for problems related to secondary storage management with an understanding of file systems and disk scheduling. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L		L									
CO2	M		M									
CO3	S		S									
CO4	S	M	S									
CO5	S	M	S									
CO6	S	M	S									

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	10	10	10
Understand	40	30	30	40
Apply	30	60	60	50
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are called assembler directives?
2. Explain the data structures used by the assembler.
3. Differentiate the concepts of single pass and multi pass translation.
4. Distinguish absolute loaders and relocating loaders.
5. Illustrate the concept of dynamic linking.

Course Outcome 2 (CO2):

1. Illustrate how nested macro calls are processed by macro processor.
2. Write the syntax for macro definition and macro call.
3. Explain the steps involved in the design of macro processor.
4. Compare and Contrast the properties of macro and subroutines.
5. Describe the data structures used by macro processor.

Course Outcome 3 (CO3):

1. Using system calls, a program in either C or C++ that reads data from one file and copies it to another file.
2. List the three major activities of an operating system in regard to memory management?
3. State the purpose of system calls?
4. List the various Memory Management schemes?
5. Distinguish between logical and physical address space?
6. Explain internal and external fragmentation of main memory?

Course Outcome 4 (CO4):

1. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

i	T(pi)
0	80
1	20
2	10
3	20
4	50

   a. Suppose a system uses FCFS scheduling. Create a Gantt chart illustrating the execution of these processes?
   b. State the turnaround time for process p3?
   c. List the average wait 0time for the processes?

2. Suppose the following jobs arrive for processing at the times indicated, each job will run the listed amount of time.

Jobs	Arrival time	Burst time (in secs.)
1	0.0	8
2	0.4	4
3	1.0	1

   Give Gantt charts illustrating the execution of these jobs using the non preemptive FCFS and SJF scheduling algorithms. Compute the average turn around time and average waiting time of each job for the above algorithms and find the best alternative.

3. Give a solution to the readers-writers problem after explaining its nature?
4. Apply Dijkstra-Haberman algorithm for deadlock avoidance?
5. Implement Simulated Semaphore(Synchronized Produced Consumer Problem)
6. Discuss how file sharing semantics of unix can be implemented. Can processing of the link and unlink commands of unix lead to deadlocks .Discuss how such deadlocks can be avoided

Course Outcome 5 (CO5):

1. Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the First-fit, Best-fit, and Worst-fit algorithms place processes of 212K, 417K, 112K, and 426K (in order)? Which algorithm makes the most efficient use of memory?
2. Consider the following page reference string 7,0,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,2. How many page faults would occur in the case?
   a. LRU

b. FIFO

c. Optimal algorithms

assuming three, five or six frames. Note that initially all frames are empty.

3.. Assume that we have a paging system with page table stored in memory

a. If a memory reference takes 200 nanoseconds how long does a paged memory reference take?

b. If we add associative registers and 75% of all page table references are found in the associative registers, what is the effective memory reference time? Assume that finding a page table entry in the associative registers takes zero time, if the entry is there.

4. Consider a demand-paging system with the following time-measured utilizations

CPU utilization	20%
Paging disk	97.7%
Other I/O devices	5%

Which (if any) of the following will (probably) improve CPU utilization? Explain your answer.

a. Install a faster CPU.

b. Install a bigger paging disk.

c. Increase the degree of multiprogramming.

d. Decrease the degree of multiprogramming.

e. Install more main memory.

f. Install a faster hard disk or multiple controllers with multiple hard disks.

g. Add pre paging to the page fetch algorithms.

h. Increase the page size.

Course Outcome 6 (CO6):

1. Put the following disk scheduling policies in the order that will result in minimum amount of head movement. a. FCFS b. Circular scan c. Elevator algorithm

2. Suppose that a disk drive has 5000 cylinders, numbered from 0 to 4999. the drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for

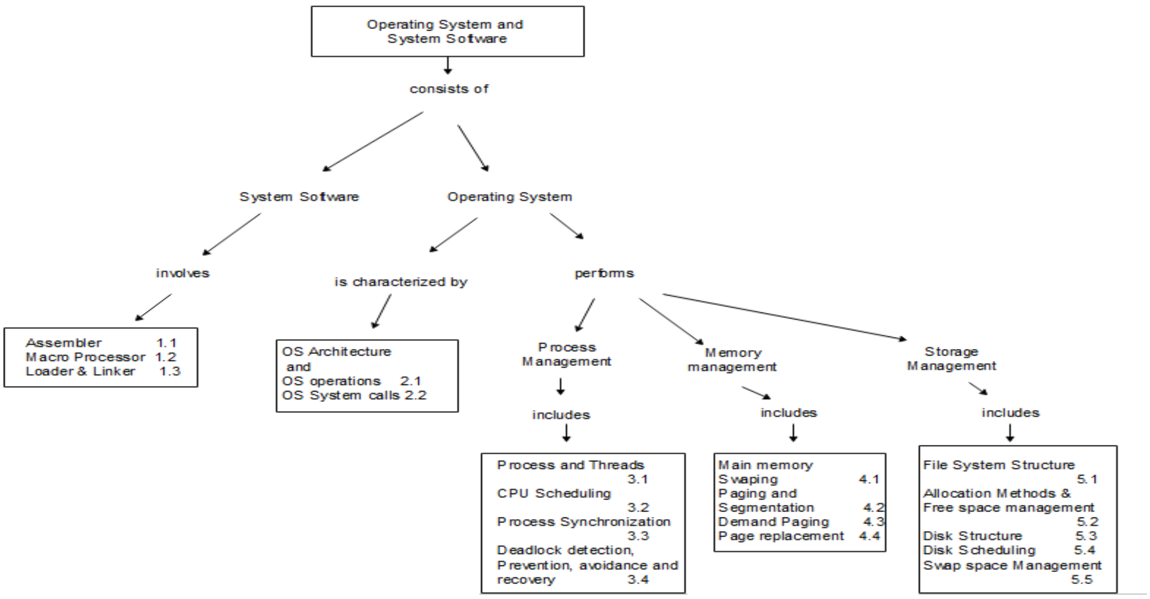
each of the disk scheduling algorithms. a. FCFS b. SSTF c. SCAN d. LOOK e. C-SCAN f. C-LOOK.

3. Explain the different type of directories in the UNIX system

4. Explain directory paths

5. Distinguish the difference between COM and EXE program

Concept Map



Syllabus

Assemblers & Macro Processors: Simple Assembly Scheme, Pass Structure of assemblers, Macro Definition and Call, Macro Expansion, Nested Macro Calls, **Linkers and Loaders:** Introduction, Relocation and linking Concepts, Types of Loaders. **Operating System Introduction & Structure:** Basics, OS Architecture, OS Operations, System calls. **Process Management:** Processes, CPU Scheduling, Process synchronization, Deadlock Detection, Deadlock Prevention, Deadlock Avoidance, Deadlock Recovery. **Memory Management:** Main Memory – swapping, Paging, Segmentation, Virtual Memory – Demand paging, Page Replacement, **Storage Management:** File System structure, Allocation methods, free space management, Disk Structure, Disk Scheduling, Swap-Space Management.

Text Books

1. D.M.Dhamdhere : System Programming, Tata McGraw Hill, 2011.
2. Avi Silberschatz, Peter Baer Galvin and Greg Gagne: Operating System Concepts, Seventh edition, John Wiley and Sons, 2006.

Reference Books

1. Leland L.Beck, System Software – An Introduction to System Programming, Pearson Education, Third Edition, 2011.
2. Andrew S. Tanenbaum, Albert S.WoodHull: Operating Systems, Design and Implementation, Third Edition, Prentice Hall, 2006.
3. William Stallings: Operating Systems: Internals and Design Principles, Fifth Edition, Prentice Hall, 2004.

Course Contents and Lecture Schedule

Module No	Topic	No. of Lectures
1	**System Software**	
1.1	Assembler -Simple Assembly Scheme Concepts	1
1.2	Macro Processors **-** Macro Definition and Call Concepts	1
1.3	Linkers and Loaders - Relocation and linking Concepts	2
2	**Operating System Introduction & Structure**	
2.1	OS Architecture and Operations	2
2.2	OS System Calls, OS Structure	2
3	**Process Management**	
3.1	Processes and Threads	2
3.2	CPU Scheduling	2
3.3	Process synchronization	2
3.4	Deadlock Detection, Prevention, Avoidance, Recovery	4
4	**Memory Management**	
4.1	Main Memory – swapping	2
4.2	Paging and Segmentation	3
4.3	Virtual Memory – Demand paging	2
4.4	Page Replacement	2
5	**Storage Management**	
5.1	File System Structure	2
5.2	Allocation methods and free space management	2
5.3	Disk Structure	2
5.4	Disk Scheduling	2
5.5	Swap-Space Management	1
	Total No of Hours	36

Course Designers:

1. Dr.P.Chitra pccse@tce.edu
2. Dr.K.Sundarakantham kskcse@tce.edu

14CS430	DESIGN AND ANALYSIS OF ALGORITHMS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This subject introduces students to the design and analysis of algorithms. On completion of this course students will be able to:

i) Demonstrate familiarity with major algorithms

ii) Determine the asymptotic time complexity of algorithms

iii) Construct efficient algorithms for solving engineering problems by using appropriate algorithm design paradigms and data structures.

Prerequisite

- 14CS270 : Problem Solving using Computers
- 14CS350 : Data Structures
- 14CS380 : Data Structures Lab

Course Outcomes

On the successful completion of the course, students will be able to

Analyze the correctness of algorithms using induction and loop invariants. (CO1)	Analyze
Analyze the worst-case, best-case and average-case running time of algorithms using asymptotics (CO2)	Analyze
Analyze the performance of a sequence of operations using amortized analysis techniques like potential method and accounting method. (CO3)	Analyze
Construct algorithms using design paradigms like divide and conquer, greedy and dynamic programming for a given problem. (CO4)	Apply
Infer when a design scenario requires the application of the different algorithm design paradigms. (CO5)	Analyze
Analyze how the performance of an algorithm is affected based on the choice of data structures the algorithm uses. (CO6)	Analyze
Construct graph-based algorithms to solve engineering problems.(CO7)	Apply
Construct polynomial reductions for standard problems with an understanding of the intractable complexity classes like NP-Complete and NP-hard.(CO8)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	S	L									
CO2	S	S	L									
CO3	S	S	L									
CO4	S	M	S									
CO5	S	S	M									
CO6	S	S	M									

CO7	S	M	S									
CO8	M	M										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	10	10	10
Understand	30	20	20	20
Apply	40	40	40	50
Analyse	10	30	30	20
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome (CO1):

1. Recall the following facts, discussed in the course: in any minimum spanning tree (of a connected, weighted graph), if we remove an edge (u, v), then the two remaining trees are each MSTs on their respective sets of nodes, and the edge (u, v) is a least-weight edge crossing between those two sets. These facts inspire Professor Goldemaine to suggest the following Divide-and-Conquer algorithm for finding an MST on a graph G = (V, E): split the nodes arbitrarily into two (nearly) equal-sized sets, and recursively find MSTs on those sets. Then connect the two trees with a least-cost edge (which is found by iterating over). Investigate whether this algorithm is correct.

2. Consider the searching problem: given an array A[1…n] and a value v output an Index i such that v = A[i] or the special value ф if v does not appear in A. If the array J is sorted, we can perform a binary search: compare P with the midpoint of the array and repeat the search on one half of array, eliminating the other half from further consideration.
(a) Construct a pseudocode for binary search as a recursive procedure.
(b) Rewrite your binary search procedure in an iterative style.
(c) Formally state pre- and post-conditions for your iterative procedure. Deduce a loop invariant, and illustrate that your procedure is correct.

3. Demonstrate that Bubble Sort works by proving that
   a. It terminates.
   b. It sorts the set of numbers provided as an input.

4. Deduce a loop-invariant to prove the correctness of the selection sort algorithm.

5. Let G be a flow network with integer capacities, and let f be an integer maximum flow in G. Suppose that we increase the capacity of an arbitrary edge in G by one unit. Construct an efficient algorithm to find a maximum flow in the modified network. Investigate why your algorithm is correct.

Course Outcome (CO2):

1. Although merge sort runs in $\Theta(n \log_2 n)$ worst-case time and insertion sort runs in $\Theta(n2)$ worst-case time, the constant factors in insertion sort make it faster for small n. Thus, it makes sense to use insertion sort within merge sort when sub-problems become sufficiently small. Consider a modification to merge sort in which n/k sub-lists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined.

    a) Show that the n/k sub-lists, each of length k, can be sorted by insertion sort in $\Theta(nk)$ worst-case time.

    b) Show that the sub-lists can be merged in $\Theta(n \log_2 (n/k))$ worst-case time.

2. Let f(n) and g(n) be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that max(f(n), g(n)) = Θ(f(n) + g(n)).

3. The following code-fragment implements Horner's rule for evaluating a polynomial $P(x) = \sum_{k=0}^{n} a_k x^k$

```
y = 0
for i = n down to 0
    y = a_i + x * y
```

   a) In terms of Θ notation, calculate the running time of this code fragment for Horner's rule

   b) Construct a pseudo-code to implement the naive polynomial-evaluation algorithm that computes each term of the polynomial from scratch. Calculate the running time of this algorithm. Compare it to the Horner's rule.

4. Investigate whether the following statement is true:
   "k ln k = Θ(n) implies k = Θ(n/ln n)"

5. Sequence the following functions by their order of growth. i.e. Deduce an arrangement g1, g2, g3, ..., g30 of the functions satisfying g1 = Ω(g2), g2 = Ω(g3),...,g29 = Ω(g30). Group your list into equivalence classes such that functions f(n) and g(n) are in the same class if and only if f(n) = Θ(g(n)).

$\lg(\lg^* n)$	$2^{\lg^* n}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$
$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$\lg^* n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	$(n+1)!$	$\sqrt{\lg n}$
$\lg^*(\lg n)$	$2^{\sqrt{2 \lg n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

Course Outcome (CO3):

1. If the set of stack operations included a MULTIPUSH operation, which pushes k items onto the stack, Investigate whether the O(1) bound on the amortized cost of stack operations continue to hold.

2. Show that if a DECREMENT operation were included in the k-bit counter example, n operations could cost as much as Θ(nk) time.

3. Suppose we perform a sequence of n operations on a data structure in which the i th operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to find the amortized cost per operation.

4. Suppose we perform a sequence of stack operations on a stack whose size never exceeds k. After every k operations, we make a copy of the entire stack for backup purposes. Show that the cost of n stack operations, including copying the stack, is O(n) by assigning suitable amortized costs to the various stack operations.

5. Analyze how to implement a queue with two ordinary stacks so that the amortized cost of each ENQUEUE and each DEQUEUE operation is O(1)

Course Outcome (CO4):

1. Let A[1 …. n] be an array of n distinct numbers. If i < j and A[i] > A[j], then the pair (i, j) is called an inversion of A.

a) Determine the inversions of the array [2, 3, 8, 6, 1]
b) Determine what array with elements from the set {1, 2, . . . , n} has the most inversions?
c) Determine the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer.
d) Determine an algorithm that determines the number of inversions in any permutation on n elements in Θ(n lg n) worst-case time.

2. Defend that the greedy algorithm that solves the scheduling problem with the goal of minimizing the time spent by the customers in a system is optimal, if it adopts the following greedy strategy:
"At each step add to the end of the schedule the customer requiring the least service time among those who remain"

3. Suppose that you have a median(S) algorithm, which finds the median element in the sequence S in O(n) time. Using this median(S) algorithm, construct a simple O(n) – linear time algorithm that solves the selection problem Select(S, k).

4. Suppose instead of running Dijkstra's algorithm till the priority queue 'Q' becomes empty, we run it as long as |Q| > 1. This change will cause the 'while' loop in Dijkstra's algorithm to execute |V| - 1 times instead of |V| times. Analyze whether the proposed algorithm is correct.

5. Suppose that you are given an $n \times n$ checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rules. At each step you may move the checker to one of three squares:

i. The square immediately above,
ii. The square that is one up and one to the left (but only if the checker is not already in the leftmost column),
iii. The square that is one up and one to the right (but only if the checker is not already in the rightmost column).

Each time you move from square x to square y, you receive $p(x, y)$ dollars. You are given $p(x,y)$ for all pairs (x, y) for which a move from x to y is legal. Do not assume that $p(x, y)$ is positive. Construct an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. Compute the running time of your algorithm as well.

Course Outcome (CO5):

1. Construct a O(n $\log_2$ n)-time algorithm that given a set S of n integers and another integer x, determines whether or not there exist two elements in S whose sum is exactly x.

2. Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write checks in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transaction ordering to check-number ordering is therefore the problem of sorting almost-sorted input. Analyze whether the procedure Insertion-sort would tend to beat the procedure Quick-sort on this problem.

3. Construct an O(n)-time algorithm by applying an appropriate design paradigm to compute the n^{th} Fibonacci number.

4. Imagine that you wish to exchange one currency for another. You realize that instead of directly exchanging one currency for another, you might be better off making a series of trades through other currencies, winding up with the currency you want. Suppose that you can trade n different currencies, numbered 1,2,… n, where you start with currency 1 and wish to wind up with currency n. You are given, for each pair of currencies i and j, an exchange rate r_{ij}, meaning that if you start with d units of currency i, you can trade for dr_{ij} units of currency j. A sequence of trades may entail a commission, which depends on the number of trades you make. Let c_k be the commission that you are charged when you make k trades. Show that, if $c_k = 0$ for all k = 1, 2,…n, then the problem of finding the best sequence of exchanges from currency 1 to currency n exhibits optimal substructure. Then show that if commissions c_k are arbitrary values, then the problem of finding the best sequence of exchanges from currency 1 to currency n does not necessarily exhibit optimal substructure.

5. Professor Stewart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend. Professor Stewart is given the tree that describes the structure of the corporation, using the left-child, right-sibling representation. Each node of the tree holds, in addition to the pointers, the name of an employee and that employee's conviviality ranking. Construct an efficient algorithm to make up a guest list that maximizes the sum of the conviviality ratings of the guests. Analyze the running time of your algorithm.

Course Outcome (CO6):

1. Illustrate the use of Fibonacci heaps in improving the asymptotic running time of the Prim's algorithm.
2. Demonstrate the benefits of using disjoint set data structure to maintain the forest of trees generated, while running the Kruskal's algorithm.

3. We have a set of n jobs to execute, each of which takes unit time. At any time T = 1,2,... we can execute exactly one job. Job i earns us a profit gi > 0 if and only if it is executed no

later than time di. Construct a greedy algorithm to solve this scheduling problem and Investigate the possibility of using a disjoint set data structure to speed up your greedy algorithm.

4. Compare the performance of three different versions of the Dijkstra's algorithm which uses one each of the following data structure to implement the min-priority queue.

(i) Array
(ii) Binary Heaps
(iii) d-Heaps
(iv) Fibonacci Heaps

5. Show how an inverted Fibonacci Heap can be used to implement Dijkstra's algorithm in a time in O(|E| + |V| log2 |V|).

Course Outcome (CO7):

1. Show that a depth first search of an undirected graph 'G' can be used to identify the connected components of 'G' and that the depth first forest contains as many trees as 'G' has connected components. More precisely, show how to modify depth-first-search so that each vertex v is assigned an integer label cc[v] between 1 and k, where 'k' is the number of connected components of 'G' such that cc[u] = cc[v] if and only if u and v are in the same connected component.

2. We are given a directed graph G = (V, E) on which each edge (u,v) $\in$ E has an associated value r(u,v), which is a real number in the range 0 ≤ r(u,v) ≤ 1 that represents the reliability of a communication channel from vertex u to vertex v. We interpret r(u,v) as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Construct an efficient algorithm to find the most reliable path between two given vertices.

3. There are two types of professional wrestlers: "good guys" and "bad guys". Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have 'n' wrestlers and a list of 'r' pairs of wrestlers between whom there is a rivalry. Recommend a O(n + r) – time algorithm that determines whether it is possible to designate some of the wrestlers as "good" and the remainder as "bad" such that each rivalry is between a "good guy" and a "bad guy". If it is possible to produce such a designation your algorithm should produce it.

4. Determine an algorithm to detect cycles in a directed graph G(V, E) in O(|V| + |E|) time.

5. Determine how the output of the Floyd Warshall's algorithm can be used to detect the presence of a negative weight cycle in a directed weighted graph G(V, E) with a weight function w: E → R

Course Outcome (CO8):

1. Let HAMD denote the problem of deciding whether a graph G<N, A> is Hamiltonian and let TSPD denote the problem of deciding whether or not there is a tour in the graph G that begins and ends at the same node; after having visited each of the other nodes exactly once and whose total cost does not exceed a given bound L. Translate a HAMD problem instance into a TSPD problem instance.

2. Analyze whether the Travelling Sales Person problem (TSP) is decision reducible.

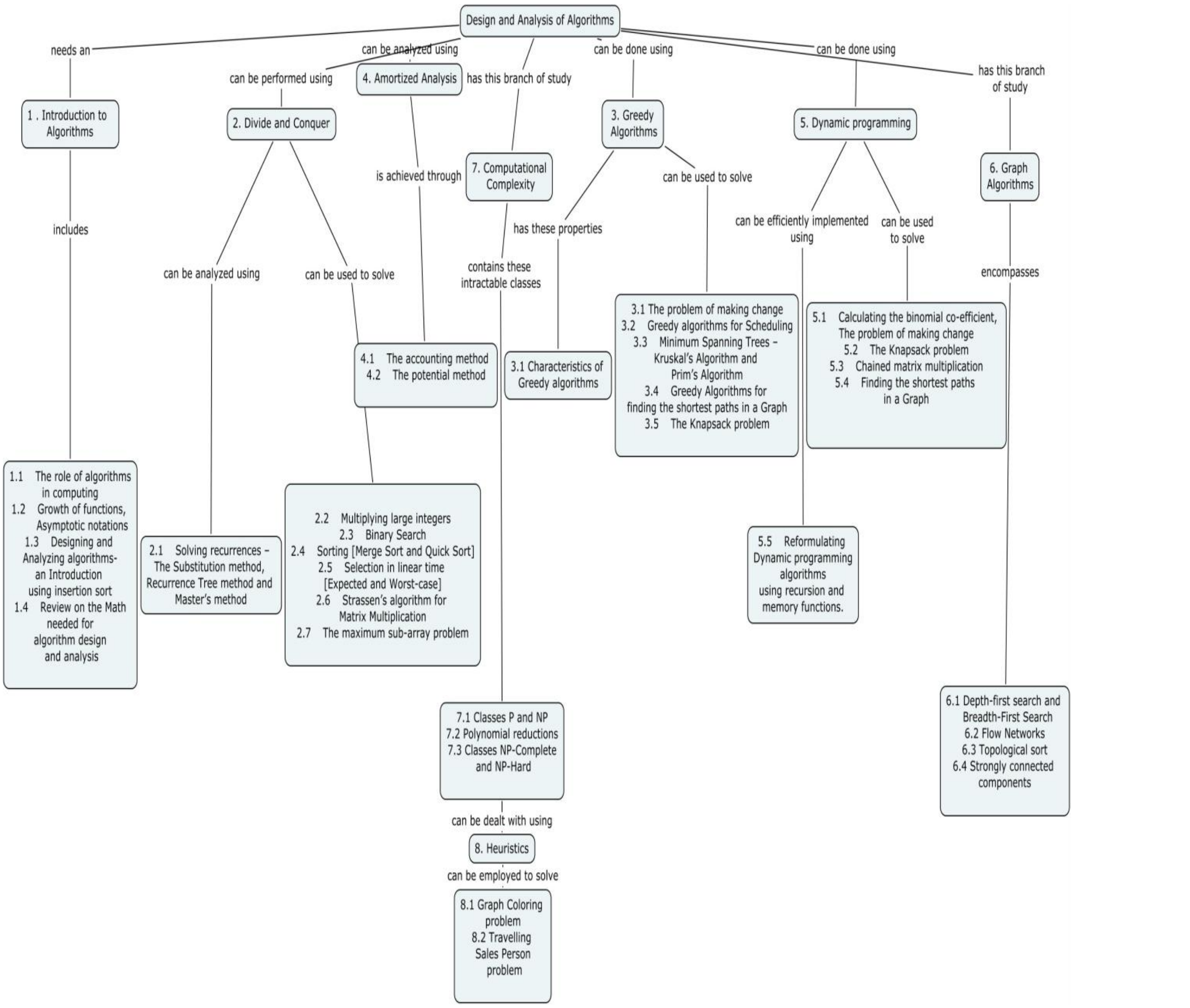
(i.e.) TSP $\equiv^{P}_{T}$ TSPD

3. Let X be an NP-Complete problem. Consider a decision problem Z Є NP such that X $\le_T^P$ Z. Then defend that Z is also NP-Complete.

4. Consider two problems A and B. Assess whether the following statements are true:
i. If A $\le_T^P$ B and if B can be solved in polynomial time, then A can also be solved in polynomial time
ii. If A and B are two decision problems such that A $\le_m^p$ B, then A $\le_T^P$ B.

5. Defend that SAT-3-CNF is NP-Complete.

Concept Map



Syllabus

Introduction to Algorithms: The role of algorithms in computing, Growth of functions, Asymptotic notations, Designing and Analyzing algorithms-an Introduction using insertion sort. Review on the Math needed for algorithm design and analysis. **Divide and Conquer**: Solving recurrences – The Substitution method, Recurrence Tree method and Master's method, Multiplying large integers, Binary Search, Sorting [Merge Sort and Quick Sort], Selection in linear time [Expected and Worst-case], Strassen's algorithm for Matrix Multiplication, The maximum sub-array problem. **Greedy Algorithms**: Characteristics of Greedy algorithms, The problem of making change, Greedy algorithms for Scheduling, Minimum Spanning Trees – Kruskal's Algorithm and Prim's Algorithm, Greedy Algorithms for finding the shortest paths in a Graph, The Knapsack problem **Amortized Analysis**: The accounting method, The potential method. **Dynamic Programming**: Calculating the binomial co-efficient, The problem of making change, The Knapsack problem, Chained matrix multiplication, Finding the shortest paths in a Graph, Reformulating Dynamic programming algorithms using recursion and memory functions. **Graph Algorithms**: Depth-first search & Breadth-First Search, Flow Networks, Topological sort, Strongly connected components **Computational Complexity**: Classes P and NP, Polynomial reductions, Classes NP-Complete and NP-Hard. **Heuristics:** Graph Coloring problem, Travelling Sales Person problem.

Text Book

1. Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest, Clifford Stein – Introduction to Algorithms, Third edition, PHI, 2010.

Reference Books

1. Gilles Brassard and Paul Bratley - Fundamentals of Algorithmics - PHI, 2000.

2. Sara Baase - Computer algorithms: Introduction to Design and Analysis –, Addison Wesley publication, 1998.

Course Contents and Lecture Schedule

Module no.	Topic	No. of lectures
1.	Introduction to Algorithms	
1.1	The role of algorithms in computing	1
1.2	Growth of functions, Asymptotic notations	2
1.3	Designing and Analyzing algorithms-an Introduction using insertion sort	1
1.4	Review on the Math needed for algorithm design and analysis	2
2.	Divide and Conquer	
2.1	Solving recurrences – The Substitution method, Recurrence Tree method and Master's method	2
2.2	Multiplying large integers	1
2.3	Binary Search	1
2.4	Sorting [Merge Sort and Quick Sort]	2
2.5	Selection in linear time [Expected and Worst-case]	1
2.6	Strassen's algorithm for Matrix Multiplication	1
2.7	The maximum sub-array problem	1
3.	Greedy Algorithms	
3.1	Characteristics of Greedy algorithms, The problem of making change	1
3.2	Greedy algorithms for Scheduling	1
3.3	Minimum Spanning Trees – Kruskal's Algorithm and Prim's Algorithm	1

3.4	Greedy Algorithms for finding the shortest paths in a Graph	1
3.5	The Knapsack problem	1
4.	Amortized Analysis	
4.1	The accounting method	1
4.2	The potential method	1
5.	Dynamic programming	
5.1	Calculating the binomial co-efficient, The problem of making change	1
5.2	The Knapsack problem	1
5.3	Chained matrix multiplication	1
5.4	Finding the shortest paths in a Graph	1
5.5	Reformulating Dynamic programming algorithms using recursion and memory functions.	1
6.	Graph Algorithms	
6.1	Depth-first search & Breadth-First Search	1
6.2	Flow Networks	1
6.3	Topological sort	1
6.4	Strongly connected components	1
7.	Computational Complexity	
7.1	Classes P and NP	1
7.2	Polynomial reductions	1
7.3	Classes NP-Complete and NP-Hard	1
8.	Heuristics	
8.1	Graph Coloring problem	1
8.2	Travelling Sales Person problem	1
	Total	36

Course Designers:

1. Mr. S. Karthick skcse@tce.edu
2. Dr. S. Padmavathi spmcse@tce.edu

14CS440	DATABASE MANAGEMENT SYSTEMS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This course aims at facilitating the student to understand the various concepts and functionalities of Database Management Systems, the method and model to store data and how to manipulate them through query languages, the effective designing of relational database and how the system manages the concurrent usage of data in multi user environment.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Develop Entity Relationship(ER) and Relational Models for a given application (CO1)	Apply
Build and manipulate relational database using Structured Query Language and relational languages.(CO2)	Apply
Develop a normalized database for a given application by incorporating various constraints like integrity and value constraints. (CO3)	Apply
Construct data structures like indexes and hash tables for the fast retrieval of data (CO4)	Apply
Illustrate different transaction and concurrency control mechanisms to preserve data consistency in a multi-user environment. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	M									
CO2	S	M	M									
CO3	S	M	M									
CO4	S		L									
CO5	S	L										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	10
Understand	20	20	40	30
Apply	60	60	40	60
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are mapping cardinalities? State their uses with examples.
2. Define weak entity set with an example.
3. Explain the concepts of generalization – specialization in E-R Model with suitable examples.
4. For the following employee database

   employee(employee-name, street, city)

   works(employee-name, company-name, salary)

   company(company-name, city)

   manages(employee-name, manager-name)

   Draw the ER diagram with all possible components and cardinalities.
5. Illustrate the concept of aggregation with suitable example.

Course Outcome 2 (CO2):

1. What do you mean by data integrity?
1. Recall the use of CHECK constraint.
2. How will you drop a table whose primary key is referenced by a foreign key in another table? Give two possible ways to accomplish this task.
3. Consider the following schema used by the Campus Book Store:

   BOOK(bookno, bookname, booktype, price)

   PUBLISHER(pname, address, phone)

   STOCK(bookno, pname, quantity)

   a) Create the tables with appropriate constraints.
   b) Alter the table PUBLISHER and change the attribute ‘address’ to composite attribute.
   c) Drop the constraint in PUBLISHER table.
4. Considering the schema structure given below

   CUSTOMER(<u>custno</u>, custname, city, phone)

   ITEM (<u>Itemno</u>, Itemname, Itemprice, QtyOnhand)

   INVOICE (<u>Invno</u> , Invdate , Custno)

   INVITEM <u>(Invno , Itemno</u> , Qty)

   For each of the following queries, give an expression in relational algebra and SQL.

   a) Find customers from ‘Chennai ‘.
   b) Display all item name along with the quantity sold.
   c) Find the customers who are not from ‘Madurai’ (use set operator)

Course Outcome 3 (CO3) :

1. Outline the desirable properties of decomposition.
2. Using the functional dependencies given A->BC , CD->E, B->D, E->A Compute B+.
3. When a relation is said to be in 1NF? Give an example.
4. Design a database for the Banking environment by following the various design phases including normalization.
5. For the following employee database

   employee(employee-name, Address)

   company(company-name, Address, setof(Phones))

   Address(street,city,pincode)

   works(employee-name, company-name, salary)

   Construct the appropriate tables by considering normalization.

Course Outcome 4 (CO4) :

1. List the advantages of dynamic hashing when compared to static hashing.
2. Stable storage can't be implemented. Explain why.
3. Consider the following account relation and construct a bitmap index on the attributes branch_name and balance, dividing balance values into 4 ranges – < 250, 250 ..<500, 500..<750 and >750.

Account_No	**Branch_Name**	**Balance**
A-217	Madurai	200
A-219	Chennai	600
A-117	Coimbatore	350
A-207	Madurai	800
A-317	Chennai	700

4. Construct a B+ tree for the following set of key values { 2,3,5,7,11,17,19,23,29,31 }
5. Suppose that we are using extendable hashing on a file that contains records with the following search key values 2,3,5,7,11,17,19,23,29,31. Show the extendable hash structure for this file, if the hash function is h(x)=x mod 8 and buckets can hold 3 records.

Course Outcome 5 (CO5) :

1. In what way can you implement atomicity in transactions? Explain.
2. Define concurrency control.
3. What do you mean by Serializability?
4. When will deadlock occur in concurrent transactions?
5. Consider the following ordering Schedule - S of transactions:

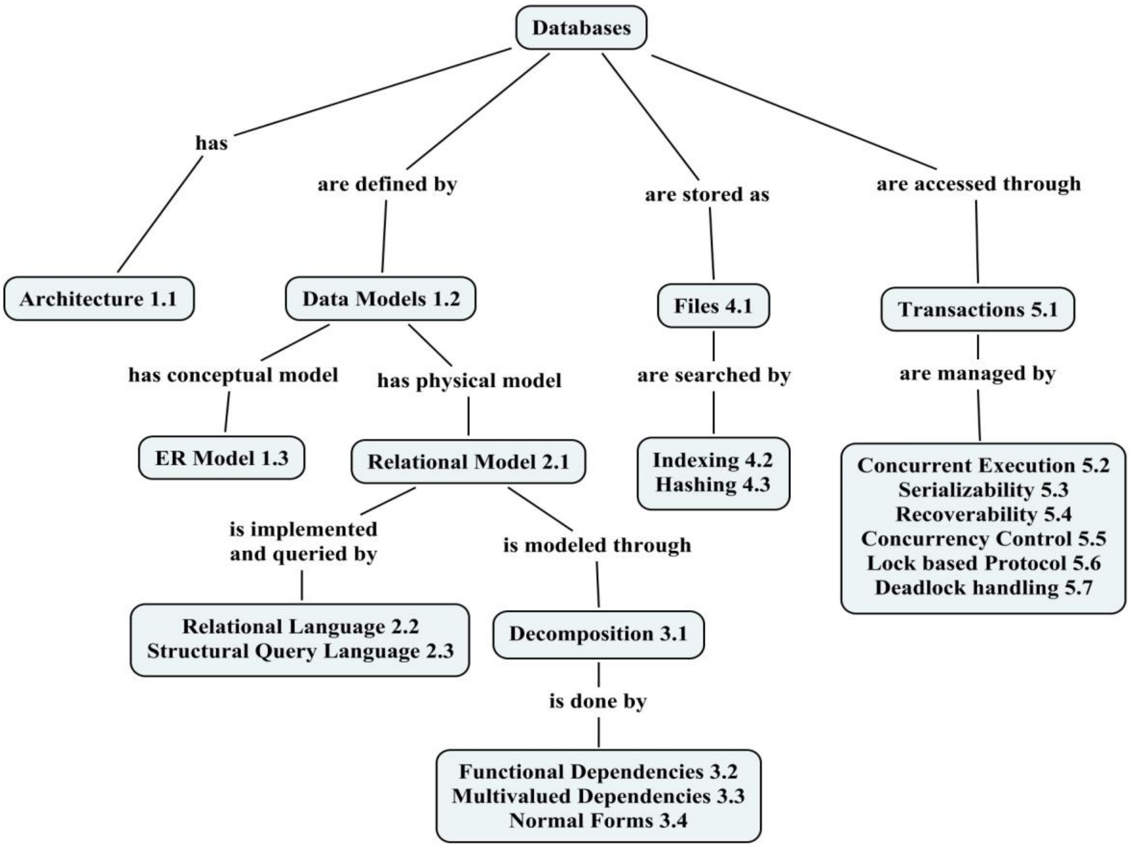
T3: W(X); T2: R(X); T3: commit; T1: W(Y); T1: commit; T2: R(Y); T2: W(Z); T2: commit; T4: R(X); T4: R(Y); T4:W(Z); T4: commit.

a) Draw the precedence graph for S.

b) Is S conflict-serializable according to the precedence graph? Justify.

c) Identify the transactions that are view equivalent and justify your answer.

Concept Map



Syllabus

Introduction to database: Purpose of database system, System Architecture, Data Models - ER data model. **Relational Databases**: Relational Model, Mapping ER model to Relational Model, Relational language, Structured Query Language - DDL, DML and TCL. **Database design**: Decomposition, Functional Dependencies, Multivalved Dependencies, and Normal forms. **Data Storage**: File Structure - Indexing, Ordered Index, Bitmap index, index files, Hashing - Static and dynamic hashing. **Transaction and Concurrency control** - Transaction concepts, Concurrent Execution, Serializability, Recoverability, Concurrency Control, Lock based protocol, Deadlock handling.

Text Book

1. Avi Silberschatz, Henry F.Korth, S.Sudarshan: “Database System Concepts”, 6th Edition, Tata McGrawHill, 2010.

Reference Books

1. Sharad Maheshwari, Ruchin Jain: “Database Management System-Complete practical Approach”, Firewall Media, Second Edition, 2006.
2. Ramez Elmasri and Shamkant B.Navathe, “ Fundamentals of Database System”, 4th edition, Pearson Education, 2004.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction to database (4)**	
1.1	Purpose of database system, System Architecture	1
1.2	Data Models – ER Model	3
2	**Relational Databases (10)**	
2.1	Relational Model, Mapping ER model to Relational Model	1
2.2	Relational language – Relational Algebra	2
2.3.1	Structured Query Language – DDL	2
2.3.2	Structured Query Language – DML	4
2.3.3	Structured Query Language – TCL	1
3	**Database design (7)**	
3.1	Decomposition	1
3.2	Functional Dependencies	2
3.2	Multivalved Dependencies	2
3.4	Normal forms	2
4	**Data Storage (7)**	
4.1	File Structure	1
4.2	Indexing, Ordered Index, Bitmap index, index files	3
4.3	Static and dynamic hashing	3
5	**Transaction and Concurrency control (8)**	
5.1	Transaction concepts	1
5.2	Concurrent Execution	1
5.3	Serializability	2
5.4	Recoverability	1
5.5	Concurrency Control	1

Module No.	Topic	No. of Lectures
5.6	Lock based protocol	1
5.7	Deadlock handling	1
	Total	**36**

Course Designers:

1. Dr. C.Deisy cdcse@tce.edu
2. Mrs. A.M.Rajeswari amrcse@tce.edu

		Category	L	T	P	Credit
14CS450	**COMMUNICATION ENGINEERING**	PC	3	0	0	3

Preamble

The Communication engineering course provides the basic knowledge about the need for modulation in communication and explains types of modulation. The course will also provide the details of electromagnetic wave propagation mechanisms and brief about the cellular concept.

Prerequisite

14CS230 - Digital circuits.
14CS310 – Probability and Statistics

Course Outcomes

On the successful completion of the course, students will be able to

Identify the Bandwidth requirements for analog modulation with an understanding of modulation techniques and modulation index. (CO1)	Apply
Illustrate the different digital modulation techniques with an understanding of data rate and bandwidth considerations. (CO2)	Apply
Compute the data rate and bandwidth requirements for different digital transmission techniques like Pulse Code Modulation, Differential Pulse Code Modulation, Delta Modulation and Adaptive Delta Modulation (CO3)	Apply
Explain the radio propagation mechanisms of Electro Magnatic wave(C04)	Understand
Illustrate the concepts like frequency reuse, cell management and call processing in cellular communication networks (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M										
CO2	S	M										
CO3	S	M										
CO4	M											
CO5	M	L										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	30	40	40	40
Apply	40	40	40	40
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define Amplitude modulation.(Remember)
2. State Carson's rule for determining the bandwidth of angle modulated wave. (Understand)
3. For an FM modulator with 40 Khz frequency deviation and a modulating signal frequency fm=10 Khz, determine the bandwidth using both the Bessel table and Carson's rule. (Apply)

Course Outcome 2 (CO2):

1. Define Information capacity.(Remember)
2. Explain the significance of I and Q channel in a QPSK modulator . (Understand)
3. Determine the maximum bit rate for an FSK signal with a mark frequency of 48 Khz, a space frequency of 52 Khz and available bandwidth of 10 Khz. (Apply)

Course Outcome 3 (CO3):

1. Define Sampling theorem.(Remember)
2. Explain the working principle of PCM . (Understand)
3. Determine the minimum number of bits required in a PCM code for a dynamic range of 80 dB. Calculate the coding efficiency. (Apply)

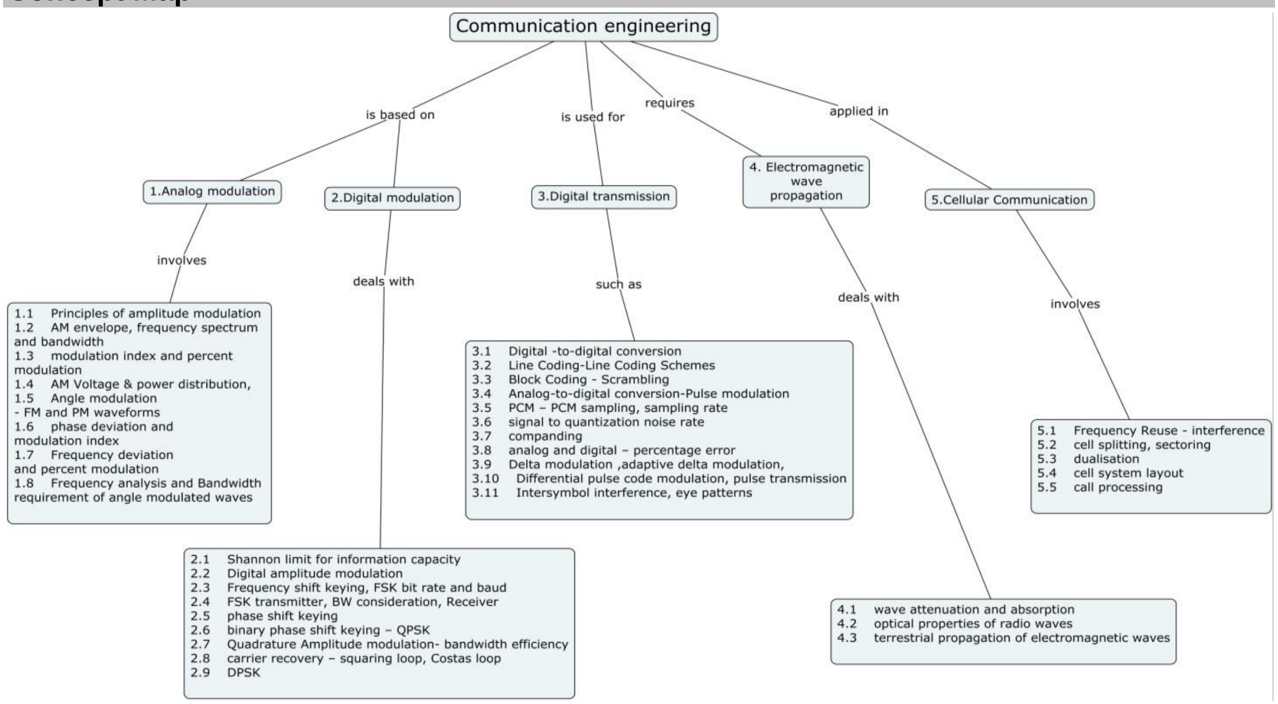
Course Outcome 4 (CO4):

1. Define Diffraction.(Remember)
2. Explain the different categories of EM wave propagation. (Understand)
3. State the applications of scattering. (Remember)

Course Outcome 5 (CO5):

1. State the different types of interference.(Remember)
2. Explain the call processing in cellular network with suitable illustrations . (Understand)
3. Determine the number of full duplex channels available in a cluster and the total capacity for a cellular system where there are 10 clusters , each consisting of 20 cells and 18 channels in each cell. (Apply)

Concept Map



Syllabus

Analog modulation- Principles of amplitude modulation, AM envelope, frequency spectrum and bandwidth, modulation index and percent modulation, AM Voltage distribution, AM power distribution, Angle modulation - FM and PM waveforms, phase deviation and modulation index, frequency deviation and percent modulation, Frequency analysis of angle modulated waves. Bandwidth requirements for Angle modulated waves

Digital modulation- Shannon limit for information capacity, digital amplitude modulation, frequency shift keying, FSK bit rate and baud, FSK transmitter, BW consideration of FSK, FSK receiver, phase shift keying – binary phase shift keying – QPSK, Quadrature Amplitude modulation, bandwidth efficiency, carrier recovery – squaring loop, Costas loop, DPSK.

Digital transmission, Digital -to-digital conversion- Line Coding - Line Coding Schemes -Block Coding - Scrambling - Analog-to-digital conversion-Pulse modulation, PCM – PCM sampling, sampling rate, signal to quantization noise rate, companding – analog and digital – percentage error, delta modulation, adaptive delta modulation, differential pulse code modulation, pulse transmission – Intersymbol interference, eye patterns.

Electromagnetic wave propagation-wave attenuation and absorption, optical properties of radio waves, terrestrial propagation of electromagnetic waves

Cellular Communication concept - Frequency Reuse - interference, cell splitting, sectoring, segmentation and dualisation, cell system layout ,call processing

Text Books

1. Wayne Tomasi, “Advanced Electronic Communication Systems”, 5/e,Pearson Education, 2007.
2. Data Communications and Networking, 5th Edition, Behrouz Forouzan, Mc Graw Hill, 2013

Reference Books

1. Simon Haykin, “Communication Systems”, 4th Edition, John Wiley & Sons. 2001.
2. H.Taub,D L Schilling ,G Saha ,”Principles of Communication”3/e,2007.
3. B.P.Lathi,”Modern Analog And Digital Communication systems”, 3/e, Oxford University Press, 2007
4. Blake, “Electronic Communication Systems”, Thomson Delmar Publications, 2002.
5. Martin S.Roden, “Analog and Digital Communication System”, 3rd Edition, PHI,
6. B.Sklar,”Digital Communication Fundamentals and Applications”2/e Pearson Education 2007.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Analog modulation**	
1.1	Principles of amplitude modulation	1
1.2	AM envelope, frequency spectrum and bandwidth	1
1.3	modulation index and percent modulation	1
1.4	AM Voltage distribution, AM power distribution	1
1.5	Angle modulation - FM and PM waveforms	1
1.6	phase deviation and modulation index	1
1.7	Frequency deviation and percent modulation	1
1.8	Frequency analysis and Bandwidth requirement of angle	2

Module No.	Topic	No. of Lectures
	modulated waves	
2.	**Digital modulation**	
2.1	Shannon limit for information capacity	1
2.2	Digital amplitude modulation	
2.3	Frequency shift keying, FSK bit rate and baud	1
2.4	FSK transmitter, BW consideration, Receiver	2
2.5	phase shift keying	1
2.6	binary phase shift keying – QPSK	1
2.7	Quadrature Amplitude modulation- bandwidth efficiency	1
2.8	carrier recovery – squaring loop, Costas loop	1
2.9	DPSK	1
3	**Digital transmission**	
3.1	Digital -to-digital conversion	1
3.2	Line Coding-Line Coding Schemes	
3.3	Block Coding - Scrambling	1
3.4	Analog-to-digital conversion-Pulse modulation	
3.5	PCM – PCM sampling, sampling rate	1
3.6	signal to quantization noise rate	1
3.7	companding	1
3.8	analog and digital – percentage error	1
3.9	Delta modulation ,adaptive delta modulation,	2
3.10	Differential pulse code modulation, pulse transmission	1
3.11	Intersymbol interference, eye patterns	1
4	**Electromagnetic wave propagation**	
4.1	wave attenuation and absorption	1
4.2	optical properties of radio waves	1
4.3	terrestrial propagation of electromagnetic waves	2
5	**Cellular Communication Concept**	
5.1	Frequency Reuse - interference	1
5.2	cell splitting, sectoring	1
5.3	dualisation	
5.4	cell system layout	1
5.5	call processing	1
	Total	**36**

Course Designers:

1. Mr. C.Senthilkumar cskcse@tce.edu
2. Mrs. S.Sridevi sridevi@tce.edu

14CS470	**PROFESSIONAL COMMUNICATION**	Category	L	T	P	Credit
		HSS	1	0	2	2

Preamble

This course provides opportunities to students to develop and demonstrate basic communication skills in technical, professional and social contexts effectively.

Prerequisite

14EG140 : English Communication

Course Outcomes

On the successful completion of the course, students will be able to

Plan, organize, write and present project reports and technical papers in the frame of the scientific method.(CO1)	Apply
Establish themselves through communication skills in corporate environment. (CO2)	Apply
Solve verbal aptitude questions relevant to placement and higher studies. (CO3)	Apply
Apply their interpersonal skills in technical, professional and social contexts. (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.									M	M		
CO2.									M	M		
CO3.									M	M		
CO4.									M	M		

S- Strong; M-Medium; L-Low

Assessment Pattern

Internal

No Continuous Assessment Test (CAT) will be conducted.

Project Report Preparation and

Technical Presentation through PPT	-	15
Listening Test	-	10
Spoken task – Group Discussion / Mock Job Interview	-	10
Writing – Verbal Aptitude for Placement and Higher studies	-	15
(The test will be conducted for 50 marks and reduced to 15)		

External(Practical)

Listening	-	20
Group Discussion	-	25
Personal Interview/Situational Conversation	-	25
Technical Presentation	-	20
Resume	-	10

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Hours	
		Theory	Practical
1	Literature Survey / Project Title Selection	1	
2	Characteristics of Technical Paper and Project Report	1	
3	Abstract / Data Presentation	1	
4	Common Errors in Technical Writing	1	
5	Bibliography and References	1	
6	Vocabulary Development	1	
7	Sentence Completion	1	
8	Error Spotting	1	
9	Interpretation of Verbal Analogy	1	
10	Interpretation of Reading (Comprehension – Conception)	1	
11	Interpretation of Reading (Comprehension – Reasoning)	1	
12	Practice for writing E-mails	1	
13	PPT Preparation / Demonstration of Technical Presentation		4
14	Preparation of Resume		2
15	Preparation for Job Interviews		4
16	Demonstration of Group Discussion Skills		4
17	Developing Listening skill (Comprehension)		3
18	Practice for Short Speeches / Situational Conversation		4
19	Development of Employability Skills		2
20	Non-Verbal Communication		1
	Total Hours	**12**	**24**

Reference Books:

1. Courseware on **"Technical Communication for Scientists and Engineers"**, IIT Bombay, 2015.
2. Cappel, Annette and Sharp, Wendy, **"Cambridge English: Objective First"**, 4th Edition, CUP, New Delhi, 2013.
3. Sue Prince, Emma, **"The Advantage: The 7 soft skills you need to stay one step ahead"**, 1st edition, Pearson, 2013.
4. Cusack, Barry. **"Improve Your IELTS Listening and Speaking Skills (With CD)"** Paperback, Macmillan, 2007.
5. Bates, Susan, **"TOEFL iBT Exam Paperback"** – Oxford, 2012.
6. Hart, Guy Brook, **"Cambridge English Business Benchmark"**, 2nd Edition, CUP 2014.

Course Designers:

1. Dr.S.Rajaram sreng@tce.edu
2. Dr.A.Tamilselvi tamilselvi@tce.edu
3. Mr.Vinoth.R vino@tce.edu
4. Ms.R.K.Jaishree Karthiga jai@tce.edu

		Category	L	T	P	Credit
14CS480	**SYSTEM SOFTWARE AND OPERATING SYSTEMS LAB**	PC	0	0	1	1

Preamble

This laboratory enables the students clearly understand the concepts of system software. Also students can implement the scheduling, process and memory management techniques.

Prerequisite

- 14CS240 : Computer Organization and Microprocessor
- 14CS270: Problem Solving using Computers
- 14CS320: Theory and Design of Programming Languages

Course Outcomes

On the successful completion of the course, students will be able to

Implement system software components like symbol table generator, single-pass and multi-pass assemblers (CO1)	Apply
Implement and demonstrate the function of system software like loaders, linkers in program execution. (CO2)	Apply
Implement page replacement algorithms and dynamic storage allocation algorithms. utilize the first fit and best fit algorithms for allocating and managing memory (CO3)	Apply
Implement scheduling algorithms such as FCFS, SJFS and Round Robin to schedule a given set of processes.(CO4)	Apply
Implement Banker's algorithm for deadlock avoidance (CO5)	Apply
Implement disk scheduling algorithms like FCFS, SSTF, SCAN and C-SCAN. (CO6)	Apply
Construct programs to demonstrate inter-process communication using shared memory, pipes and message queues. (CO7)	Apply
Implement solutions to the critical section problem using semaphores.(CO8)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S		M		M				M	L		
CO2	S		M		M				M	L		
CO3	S		M		M				M	L		
CO4	S		M		M				M	L		
CO5	S		M		M				M	L		
CO6	S		M		M				M	L		
CO7	S		M		M				M	L		
CO8	S		M		M				M	L		

S- Strong; M-Medium; L-Low

List of Experiments

1. Implementation of symbol table.
2. Implementation of Single pass assembler.
3. Implementations of Multi pass assembler.
4. Implementation of Linker, absolute and relocatable Loader.
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
6. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
7. Developing Application using Inter Process Communication (using shared memory, pipes or message queues)
8. Implement Bankers Algorithm for Deadlock Avoidance
9. Implement the Producer – Consumer problem using semaphores (using UNIX system calls).
10. Implement First fit and Best fit memory management schemes
11. Implement First come first serve and LRU Page replacement Algorithms
12. Implement Disk management using Algorithms such as FCFS,SSTF,SCAN and C-SCAN

Course Designers

1. Dr. K.Sundarakantham kskcse@tce.edu
2. Mr. K.Narasimha Mallikarjunan arjunkambaraj@tce.edu
3. Mrs. B.Subbulakshmi bscse@tce.edu

15CS490	ALGORITHMS LAB	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

The objective of this laboratory course is to enable students to solve algorithmic problems by appropriately modeling the problem, choosing and/or designing efficient data structures and algorithms to meet the problem constraints and implementing the algorithm in C/C++.

Prerequisites

14CS270 : Problem Solving using Computers

14CS350 : Data Structures and Algorithms

14CS380 : Data Structures Lab

Course Outcomes

On the successful completion of this course, students will be able to

CO1: Develop efficient algorithms and implementation schemes for solving a given problem using appropriate data structures and design techniques like divide and conquer, greedy, branch and bound and dynamic programming.	Apply
CO2: Compare the suitability of several candidate data structures, algorithms and implementation schemes to solve a problem, based on the time, space complexities and problem constraints imposed.	Analyze
CO3: Model, implement and evaluate the algorithms designed using a high-level programming language.	Analyze

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	S	S	M	M			M	S	M		M
CO2	S	S	S	M	M			M	S	M		M
CO3	S	S	S	M	M			M	S	M		M

S- Strong; M-Medium; L-Low

Course Level Assessment Questions

Course Outcomes (CO1, CO2, CO3):

1. You are given an array A[ ] of n positive integers, and a target sum t (again a positive integer). Your task is to find a nonempty sub-array A[i…j] such that t = A[i] + A[i + 1] + … + A[j], or report that no such sub-array exists. Write an O(n)-time program to solve this problem.

2. A college is building a complex of classrooms for staging several events like classes, labs, quizzes, conferences, and so on. It has a complete list of events before it starts its design process. Each event is specified by an interval (a, b), where a is the start time of the event, and b its finish time. We assume that the events have an hourly schedule, that is, a and b are integers. We also assume that a < b for each event. The intervals are assumed to be open, that is, two intervals like (a, b) and (b, c) are considered non-overlapping. For example, you can schedule the events (1, 3) and (3, 6) in the same classroom. Moreover, multiple events with the same start and finish times may be present. For instance, you may

have multiple classes running during the same period (5, 6). The college administration plans to design a complex such that all the events can be scheduled with as few classrooms as possible. Write a function minClassRoomCount() to solve this problem, that is, a function to return the minimum possible number of classrooms needed so that all the events can be scheduled. This function should run in $O(n(\log n + t))$ time (or better), where n is the number of events, and t is the count returned by the function.

3. Ms. Zyzzyva receives m items in her office at Fast Courier Transform Private Limited. Each item has a cost (or weight). She needs to put the items in cartons for shipping to a foreign country. She must put the items in the same sequence as she has received them. Moreover, the total cost of the items packed in each carton must not exceed a capacity bound C (this may be the weight limit of a carton or a restriction on total cost imposed by the customs department of the destination country). In order to reduce packing and shipping costs, Ms. Zyzzyva needs to minimize the number of cartons to pack all the m items without violating the sequence and capacity constraints. Let the item costs and C be integers. Treat C as a constant. For this problem, take C = 100. Finally, assume that each item cost is ≤ C (otherwise, the problem has no solution). We can restate Ms. Zyzzyva's problem as follows.

Let $A = [a_0, a_1, \dots , a_{m-1}]$ be an array of m integers. Decompose the array into a concatenation of k subarrays ($A = A_1A_2 \dots A_k$) such that no Ai is empty, and k is as small as possible.

Part 1: In this part, assume that each ai is positive. Write a function packpos() to solve Ms. Zyzzyva's problem in this case. Your function should implement a greedy algorithm, and run in O(m) time.

Part 2 : Now, allow the array elements ai to be negative, that is, A[] is now a mixture of positive and negative integers. (If you need a physical interpretation, think of a helium-filled balloon having “negative” weight or a discount coupon having negative cost.) Write a function packposneg() to solve the packing problem. Your function should run in $O(m^2)$ time. Use dynamic programming. Write a comment before the function, clearly stating what recurrence relation you are using in your dynamic-programming algorithm.

Part 3 Finally, suppose that there are two arrays $A = [a_0, a_1, \dots , a_{m-1}]$ and $B = [b_0, b_1, \dots , b_{n-1}]$. For simplicity, assume again that all a_i and b_j are positive integers. These two arrays stand for two different types of items (one from multi-national companies, the other from ulti-national companies). Ms. Zyzzyva may mix items of two types in a carton. But the total cost of the items in each carton must be ≤ C, and the items of both types must be packed in the sequence they appear in the respective arrays. In other words, she needs a simultaneous decomposition of the two arrays as $A = A_1A_2 \dots A_k$ and $B = B_1B_2 \dots B_k$ (k is the same for both the arrays) such that for each i, either A_i or B_i (or both) is/are non-empty, and the sum of the elements of A_i and B_i is ≤ C. The goal is to make k as small as possible. Write an O(mn)-time function packtwo() to solve the packing problem for two arrays. This function too should take a dynamic-programming approach. Write a comment before the function, clearly stating what recurrence relation you are using in your dynamic-programming algorithm.

Write a main() function to do the following:
• Read m and an array A of m positive integers. Call the function of Part 1 to optimally pack the items in A. Print which items are packed in each carton. Do not rewrite A. We need it again at the end.
• Read a second array A' of a mixture of m positive and negative integers. Call the function of Part 2 to optimally pack the items in A'. Print which items are packed in each carton.
• Read n and an array B of n positive integers. Call the function of Part 3 to optimally pack the items in A and B. Print which items are packed in each carton.

Submit a single C/C++ file solving all the parts. Do not use global/static variables.

List of Experiments

Problem statements that require the student to make use of each of the following data structure and algorithm design technique have to be provided. Students will be expected to understand and model the problem, design efficient algorithms, implement the algorithms designed using C/C++ and refactor the design/code after testing the algorithm against the test cases provided.

1. Basic algorithm design and asymptotic notations.
2. Algorithm design involving sorting and selection.
3. Algorithm design involving union find structures.
4. Algorithm design involving hash tables and priority queues.
5. Algorithm design involving trees like search trees, interval trees, k-d trees and AVL trees.
6. Divide and Conquer Algorithm design.
7. Greedy Algorithm design.
8. Dynamic Programming Algorithm design.
9. Branch-and-Bound Algorithms.
10. Algorithm design involving graphs.
11. Algorithm design involving String matching.
12. Algorithm design based on Network Flows.

Course Designer:

1. Mr. S. Karthick skcse@tce.edu

14CS4C2	CAPSTONE COURSE - I

Category	L	T	P	Credit
PC	0	0	2*	2

Preamble

The purpose of this course is to apply the concept of engineering fundamentals and an engineering specialization to solve complex engineering problems.

Syllabus

Engineering Group1

Digital Logic:
Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).

Computer Organization:
Machine instructions and addressing modes. ALU, Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).
.

Engineering Group 2

Programming and Data Structures:
Programming in C. Recursion. Arrays, Pointers, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.

Algorithms:
Searching, sorting, hashing. Asymptotic worst case time and space complexity

Databases:
ER-model. Relational model: relational algebra, tuple calculus, Integrity constraints.

Assessment Pattern

(Common to B.E./B.Tech Programmes)

Test 1: Engineering Group 1 (60 Marks), Duration: 90 Minutes
Objective Type Questions : 30 (15 Questions from each course)
Fill up the blanks : 30 (15 Questions from each course)

Test 2: Engineering Group 2 (60 Marks), Duration : 90 Minutes
Objective Type Questions : 30 (10 Questions from each course)
Fill up the blanks : 30 (10 Questions from each course)

Test 3: Comprehensive (60 Marks), Duration : 90 Minutes
Objective Type Questions : 30 (15 Questions from each group)
Fill up the blanks : 30 (15 Questions from each group)

Test	Marks Obtained	Converted to
Test1	60 Marks (Max)	20 Marks (Max)
Test 2	60 Marks (Max)	20 Marks (Max)
Test 3	60 Marks (Max)	60 Marks (Max)
		100 Marks (Max)

No re-test will be conducted at any circumstances

* - 2 hours/ week is allotted for off-class practical work

Course Designers:

1. Dr. S. Padmavathi spmcse@tce.edu
2. Mrs. B. Subbulakshmi bscse@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

FIFTH SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY
(For the candidates admitted from 2016-2017)

FIFTH SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS510	Numerical Methods and Number Theory	ES	2	2	-	3
14CS521	Computer Networks	PC	3	-	-	3
14CS530	Theory of Computation	PC	3	-	-	3
14CS540	Computer Architecture	PC	3	-	-	3
14CSPX0	Program Elective - I	PE	3	-	-	3
THEORY CUM PRACTICAL						
14CS571	Software Engineering: Theory and Practice	PC	2	-	2	3
PRACTICAL						
14CS580	Databases Lab	PC	-	-	2	1
14CS590	Network Programming Lab	PC	-	-	2	1
Total			**16**	**2**	**6**	**20**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core
PE : Program Elective

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture is equivalent to 1 credit
2 Hours Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

FIFTH SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS510	Numerical Methods and Number Theory	3	50	50	100	25	50
2	14CS521	Computer Networks	3	50	50	100	25	50
3	14CS530	Theory of Computation	3	50	50	100	25	50
4	14CS540	Computer Architecture	3	50	50	100	25	50
5	14CSPX0	Program Elective - I	3	50	50	100	25	50
THEORY CUM PRACTICAL								
6	14CS571	Software Engineering: Theory and Practice	3	50	50	100	25	50
PRACTICAL								
7	14CS580	Databases Lab	3	50	50	100	25	50
8	14CS590	Network Programming Lab	3	50	50	100	25	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

14CS510	NUMERICAL METHODS AND NUMBER THEORY	Category	L	T	P	Credit
		ES	2	1	0	3

Preamble

Number theory deals with various applications such as encrypting and decrypting messages, algorithms for finding GCD of integers and concept of modular arithmetic in generating Pseudo random numbers. Numerical method deals with finding approximate solutions of polynomial, simultaneous algebraic equations, Interpolation, Differentiation and Integration, ODEs and PDEs by various Numerical techniques. The course is designed to impart the knowledge and understanding of the above concepts to computer science engineering students and apply them in their areas of specializations.

Prerequisite

14 MA110: Engineering Mathematics I

Course Outcomes

On the successful completion of the course, students will be able to

Apply the concepts of modular arithmetic in appropriate Computer Science and Engineering applications like pseudo-random number generation, encryption and decryption of messages. (CO1)	Apply
Solve algebraic and transcendental equations by applying appropriate numerical methods. (CO2)	Apply
Apply the convergence criterion for finding the positive roots of polynomial equations. (CO3)	Apply
Apply appropriate numerical methods to compute an approximate solution for simultaneous linear algebraic equations. (CO4)	Apply
Apply various methods to find the inverses of a non-singular matrix. (CO5)	Apply
Apply appropriate numerical methods to find approximate solutions for problems related to interpolation, differentiation and integration. (CO6)	Apply
Apply various predictor corrector methods for finding approximate solutions of Ordinary Differential Equations. (CO7)	Apply
Apply various computational methods for finding approximate solutions of Partial Differential Equations of different types. (CO8)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M										
CO2	S	M										
CO3	S	M										
CO4	S	M										
CO5	S	M										
CO6	S	M										
CO7	S	M										

CO8	S	M										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	10	10	10	0
Understand	30	30	30	30
Apply	60	60	60	70
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1)

1. Apply modular exponentiation to find 2^{644} modulo 645
2. Demonstrate the role of number theory in RSA encryptions end encryt the message "STOP"

Course Outcome 2 (CO2)

1. Predict the positive root of $3x-\sqrt{1+Sin\,x}=0.$ by iteration method.
2. Find the positive root of $x^3-8x-40=0.$ by Horner's method.
3. Find all the roots of $2x^3-7x^2+7x-2=0.$ by Graeffe's root Squaring method up to three squarings.

Course Outcome 3 (CO3)

1. Find the quadratic factor of the equation $x^4+5x^3+3x^2\,\_\,5x-9=0.$ close to x^2+3x-5 by Barstow's method.
2. Find the positive root of $x-Cos\,x=0.$ using Newton's method.

Course Outcome 4 (CO4)

1. Find the approximate solution to the system of equations $x+3y+10z=24$; $2x+17y+4z=35$; $28x+4y-z=32.$ by Gauss Seidel method.
2. Find the approximate solution to the system of equations $8x+y+z=8$; $2x+4y+z=4$; $x+3y+3z=5.$ by Gauss Jacobi method.

Course Outcome 5 (CO5)

1. Calculate the inverse of the matrix $\begin{pmatrix} 4 & 1 & 2 \\ 2 & 3 & -1 \\ 1 & -2 & 2 \end{pmatrix}$ by Crout's method.
2. Calculate the inverse of the matrix $\begin{pmatrix} 1 & 6 & 1 \\ 1 & 2 & 0 \\ 0 & -0 & 3 \end{pmatrix}$ by Gauss Elimination method.

Course Outcome 6 (CO6)

1. From the following data , find θ at x = 43 and x = 84

x:	40	50	60	70	80	90
θ:	184	204	226	250	276	304

2. The Population of a certain town is given below. Find the rate of growth of the population in 1931 ,1941 , 1961 and 1971

Year x:	1931	1941	1951	1961	1971
Population y: In 1000's	40.62	60.80	79.95	103.56	132.65

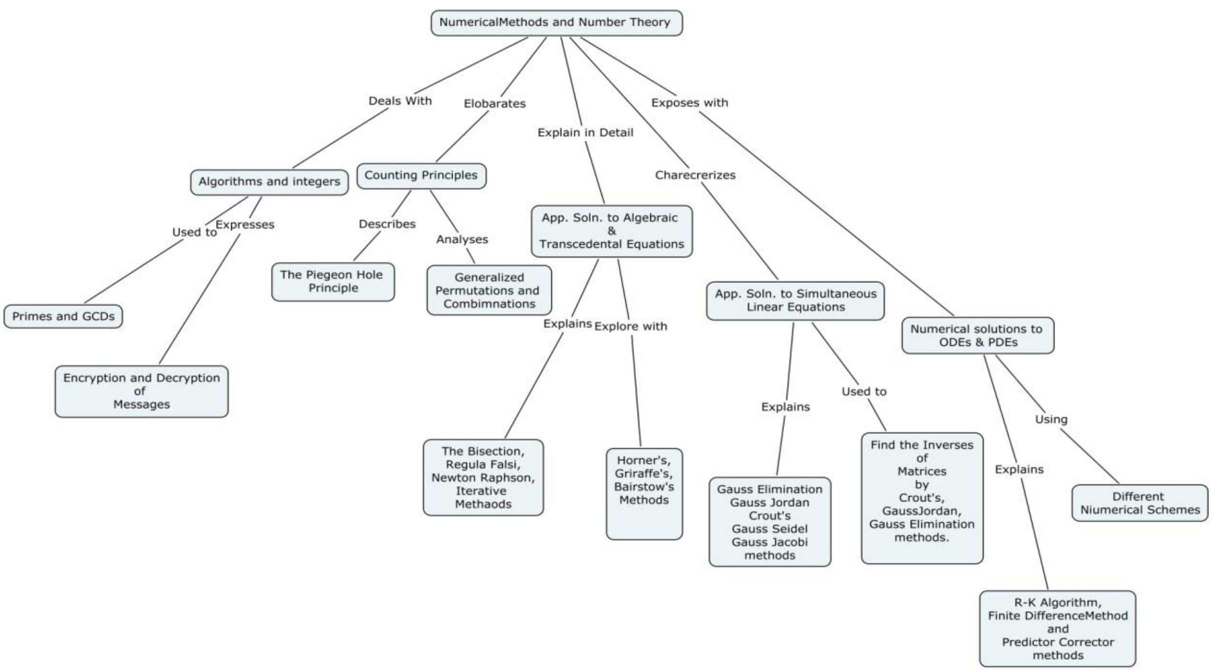
Course Outcome 7 (CO7)

1. Calculate $y(0.8)$ given that $y' = y - x^2$; $y(0)=1;\ y(0.2)=1.12; y(0.3)=1.46;\ y(0.6)=1.73;$ using Milne's predictor corrector method.
2. Calculate $y(0.8)$ given that $y' = xy + y^2$; $y(0)=1;\ y(0.2)=1.12; y(0.3)=1.46;\ y(0.6)=1.73;$ using Adam's predictor corrector method.

Course Outcome 8 (CO8)

1. Solve: $u_{xx} + u_{yy} = 0$; over the square mesh of side 4 satisfying the following boundary conditions
$u(0, y) = 0$; $0 \le y \le 4$; $u(4, y) = 12 + y$; $0 \le y \le 4$;
$u(x,0) = 3x$; $0 \le x \le 4$; $u(x,4) = x^2$; $0 \le x \le 4$.
2. Using Bender Schmidt method find the solution of the parabolic equation $\dfrac{\partial^2 u}{\partial x^2} - 2\dfrac{\partial u}{\partial t} = 0$; $where\ u(0,t) = 0 = u(4,t)$; $u(x,0) = x(4-x)$.. Assume h=1.Find the values up to t=5.

Concept Map



Syllabus

The fundamentals: Algorithms and the integers: The integers and division, Primes and Greatest Common Divisors, Integers and algorithms, Applications of Number Theory. **Solutions to Algebraic and Transcendental Equations**: Bisection, Regula falsi, Newton-Raphson, Iterative Methods, Horner's Method, Graeffe's Root Squaring Method, Bairstow's method.

Solution of Simultaneous linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Crout's, Gauss Seidel, Gauss Jacobi, Inversion by Gauss Jordan and Crout's Method, Relaxation method, Power method for finding eigen values.

Interpolation, Differentiation and Integration: Newton Gregory's forward and backward difference interpolation formulae, Gauss's and Lagrange's interpolation formulae, Newton's forward and backward formulae for derivatives, Trapezoidal, Simpson's 1/3rd and 3/8th rules

Numerical solutions of ODE and PDE: Runge Kutta Method of fourth order, Predictor-Corrector Method- Adams Bashforth, Milne's Method, Boundary Value Problem- Solution by Finite difference method, Classification of PDE – Finite difference approximations, Solution of Elliptic equations by Leibmann's method, Solution of parabolic and Hyperbolic equations by explicit methods.

Text Books

1. Jain.M.K.Iyengar.S.R.K. JainR.K., "Numerical Methods for Scientific and Engineering Computation"-Fifth edition, New Age International Publishers, New Delhi-2009.
2. Kenneth H.Rosen , "Discrete Mathematics and its Applications" – Sixth edition, Tata McGraw-Hill Publishing Company Limited- New Delhi .2012.
3. B.S.Grewal," Numerical Methods",- Nineth Edition- Khanna Publishing Company- New Delhi -2010.

Reference Books

1. Robert.J Schilling, Sandra L.Harris "Applied Numerical Methods for Engineers Using Mat lab and C" Thomson Books/cole,1999
2. Sastry S.S "Introductory Methods of Numerical Analysis" Fifth edition Prentice Hall of India , New Delhi -2006

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**The fundamentals: Algorithms and the integers**	
1.1	The integers and division	1
1.2	Primes and Greatest Common Divisors	1
	Tutorial	2
1.3	Integers and algorithms	1
1.4	Applications of Number Theory	1
	Tutorial	2
2	**Solutions to Algebraic and Transcendental equations**	
2.1	Bisection , Regula-falsi Method	1

2.2	Newton- Raphson method , Iterative method	1
	Tutorial	2
2.3	Horner's method	1
2.4	Graffe's root squaring method	1
2.5	Bairstow's method	1
	Tutorial	2
3	**Solution of simultaneous linear algebraic equations**	
3.1	Gauss elimination and Gauss Jordan methods	1
3.2	Crout's method, Gauss Jacobi and Gauss Siedal methods	2
	Tutorial	3
3.3	Inversion by Gauss Jordan and Crout's methods	1
3.4	Relaxation Method	1
3.5	Power method for finding Eigen values	1
	Tutorial	3
4	**Interpolation, Differentiation and Integration**	
4.1	Newton Gregory's forward and backward difference interpolation formulae	1
4.2	Gauss's and Lagrange's interpolation formulae	1
	Tutorial	2
4.3	Newton's forward and backward formulae for derivatives	1
4.4	Trapezoidal, Simpson's $1/3^{rd}$ and $3/8^{th}$ rules	1
	Tutorial	2
5	**Numerical Solution of ODE and PDE**	
5.1	Runge Kutta Method of fourth order	1
5.2	Predictor-Corrector Method- Adams Bash forth, Milne's Method	1
5.3	Boundary value problem – Solution by finite difference method	1
	Tutorial	3
5.4	Classification of PDE and solution of elliptic equations by Leibmann's Method	1
5.5	Solution of parabolic and Hyperbolic equations by explicit methods	1
	Tutorial	3
	TOTAL	48

Course Designers:

1. Mr. N.K.Chandrasekaran nkcmat@tce.edu
2. Mr. B.Vigneswaran bvmat@tce.edu
3. Dr.A.Anitha aamat@tce.edu

		Category	L	T	P	Credit
14CS521	**COMPUTER NETWORKS**	PC	3	0	0	3

Preamble

The course on computer networks has been organised around the layers of layered architecture in a top down manner. The course presents distinct concepts and protocols as well as the overall picture. The top down approach starts with general introduction, followed by application layer which is a high-growth area. After the initial coverage on application layer, the required network services to support the applications are presented. Thus computing the performance metrics of a packet network and the lower layer functions up to the link layer and MAC sub layer are covered.

Prerequisite

14CS230: Digital Circuits

14CS310: Probability and Statistics

14CS450: Communication Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Explain the overview of the Internet, layered architecture, data transfer through the Internet and the operation of network applications like HTTP, FTP, Email, DNS and P2P. (CO1)	Understand
Compute the performance metrics of a packet switched network. (CO2)	Apply
Analyze the performance of transport layer protocols and the beneficial effects of adopting suitable congestion control schemes. (CO3)	Analyze
Construct routing and forwarding solutions for packet switching networks, with an understanding of the underlying switching techniques, router architectures, algorithms and protocols. (CO4)	Apply
Solve flow and error control issues in the data link layer by applying appropriate flow control and error control schemes . (CO5)	Apply
Analyze the performance implications of various operational parameters of random access protocols like ALOHA, slotted ALOHA and CSMA/CD by applying the principles of link layer, MAC sub layer protocols and switches. (CO6)	Analyze

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M	L	L									
CO2.	S	M	M									
CO3.	S	S	M	L					M			L
CO4.	S	M	S									
CO5.	S	M	S									
CO6.	S	S	M	L					M			L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	50	30	20	20
Apply	30	30	40	40
Analyze	0	20	20	20
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List any 2 available residential access technologies. (Remember)
2. What is the need for layering? (Understand)
3. List the layers of TCP/IP protocol stack. (Remember)
4. Give an example of a circuit-switched network. (Remember)
5. Why is transport layer called a true end to end layer? (Understand)
6. List two non proprietary Internet applications and the application-layer protocols that they use.
   (Remember)
7. What is the difference between network architecture and application architecture?

   (Understand)
8. Suppose you wanted to do a transaction from a remote client to a server as fast as possible. Would you use UDP or TCP? Why? (Understand)
9. Why do HTTP, FTP, SMTP, and POP3 run on top of TCP rather than on UDP?
   (Understand)
10. Why is it said that FTP sends control information "out-of-band"? (Understand)
11. From a user's perspective, what is the difference between the download-and delete mode and the download-and-keep mode in POP3? (Understand)
12. List at least four different applications that are naturally suitable for P2P architectures.
    (Understand)
13. Describe the general format of HTTP request message. (Understand)
14. Explain the components of cookies and their use. (Understand)
15. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario?
    (Understand)
16. Explain the implementation of a simple database in a P2P network.(Understand)
17. Explain the operation of domain name resolution. (Understand)
18. Explain the features of IMAP. (Understand)

Course Outcome 2 (CO2):

1. Define end to end delay. . (Remember)
2. Define the term throughput of a network. (Remember)
3. Explain the various time delays encountered by packets in a datagram network.
   (Understand)
4. Explain the components of a total nodal delay. (Understand)

5. Consider sending a large file of *F* bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of *S* bits each and adds 80 bits of header to each segment, forming packets of $L = 80 + S$ bits. Each link has a transmission rate of *R* bps. Find the value of *S* that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay. (Apply)
6. Suppose two hosts, A and B, are separated by 20,000 kilo meters and are connected by a direct link of $R = 2$ Mbps. Suppose the propagation speed over the link is $2.5 * 10^8$ meters/sec.

   a. Calculate the bandwidth-delay product.

   b. Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

   c. Provide an interpretation of the bandwidth-delay product.

   d. What is the width (in meters) of a bit in the link?

   e. Derive a general expression for the width of a bit in terms of the propagation speed *s,* the transmission rate *R,* and the length of the link *m*. (Apply)

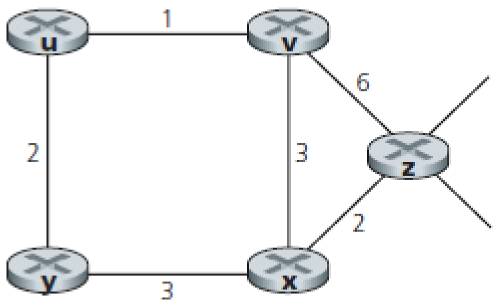
Course Outcome 3 (CO3)

1. List the features of UDP. (Remember)
2. Explain the purpose of UDP checksum calculation . (Understand)
3. Explain the operation of multiplexing and de multiplexing. (Understand)
4. What is the purpose of TCP buffers ? (Understand)
5. Explain if the UDP checksum calculation can detect all types of bit errors? (Understand)
6. Suppose Host A sends two TCP segments to Host B over a TCP connection. The first segment has a sequence number 90; the second has a sequence number 110. How much data is in the first segment? If the first segment is lost but the second segment arrives at B. What will be the acknowledgment number in the acknowledgment that Host B sends to Host A? (Understand)
7. Consider transferring an enormous file of *L* bytes from Host A to Host B. Assume an MSS of 536 bytes . What is the maximum value of *L* such that TCP sequence numbers are not exhausted? For the calculated value of L, find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously. (Analyze)
8. Consider a simplified TCP's AIMD algorithm where the congestion window size is measured in number of segments, not in bytes. In additive increase, the congestion window size increases by one segment in each RTT. In multiplicative decrease, the congestion window size decreases by half (if the result is not an integer, round down to the nearest integer). Suppose that two TCP connections, C1 and C2, share a single congested link of speed 30 segments per second. Assume that both C1 and C2 are in the congestion avoidance phase. Connection C1's RTT is 50 msec and connection C2's RTT is 100 msec. Assume that when the data rate in the link

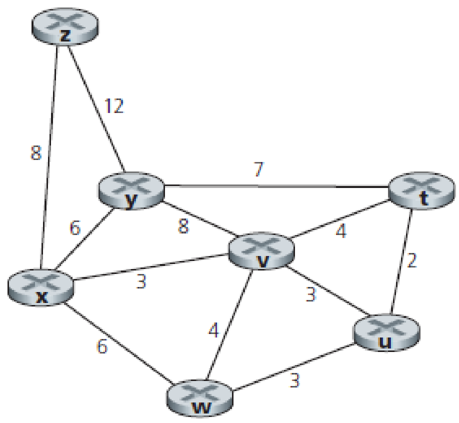
exceeds the link's speed, all TCP connections experience data segment loss.If both C1 and C2 at time t0 have a congestion window of 10 segments, what are their congestion window sizes after 1000 msec? (Analyze)

Course Outcome 4 (CO4)

1. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node *z*.. (Apply)

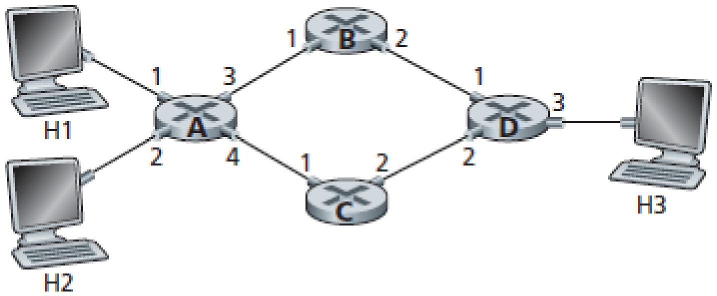


2. Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from *x* to all network nodes. (Apply)



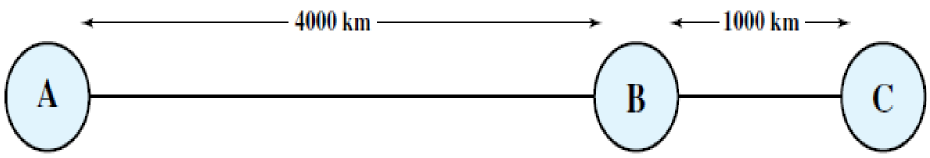
3. Draw the IPv4 header format and the functions of various fields. (Understand)
4. List the drawbacks of IPv4. (Remember)
5. List the features of services provided by IPv4. (Remember)
6. List the new features of IPv6. (Remember)
7. Discuss the count to infinity problem in distance vector routing. (Understand)
8. Explain the fragmentation of a 2400-byte datagram sent through a link with an MTU of 700 bytes. (Understand)
9. Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router. Suppose the two packets are to be sent to two *different* output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a *shared bus*? Is it possible when the fabric uses a *crossbar*? (Understand)
10. Consider the following datagram network. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.Also write down a forwarding table in router A, such that all traffic from H1 destined to host H3

is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4. (Apply)



Course Outcome 5 (CO5)

1. State the functions of link layer. (Remember)
2. Explain the concept of bit stuffing. (Understand)
3. Explain the need for flow control. (Understand)
4. List the advantage and advantage of error detection. (Remember)
5. Sixteen-bit messages are transmitted using a Hamming code, using even parity. Determine the number of check bits needed to ensure that the receiver can detect and correct single bit errors? Show the bit pattern transmitted for the message 1101001100110101. (Apply)
6. A 12-bit Hamming code whose hexadecimal value is 0xE4F arrives at a receiver. Determine the original transmitted value in hexadecimal, assuming not more than 1 bit is in error. (Apply)
7. A bit stream 10011101 is transmitted using the standard CRC method, with the generator 1001. Show the actual bit string transmitted. Suppose the third bit from the left and the second bit from the right of the transmitted message are inverted during transmission. Show that this error is detected (Apply)
8. In the following figure, frames are generated at node A and sent to node C through node B. Determine the minimum data rate required between nodes B and C so that the buffers of node B are not flooded, based on the following:
   - The data rate between A and B is 100 kbps.
   - The propagation delay is for both lines.
   - There are full duplex lines between the nodes.
   - All data frames are 1000 bits long; ACK frames are of negligible length.
   - Between A and B, a sliding-window protocol with a window size of 3 is used.
   - Between B and C, stop-and-wait is used.
   - There are no errors. (Apply)



9. A channel has a data rate of *R* bps and a propagation delay of *t* s/km. The distance between the sending and receiving nodes is *L* kilometers. Nodes exchange fixed-size frames of *B* bits. Find a formula that gives the minimum sequence field size of the frame as a function of *R, t, B,* and *L.* (considering maximum utilization). Assume that

ACK frames are negligible in size and the processing at the nodes is instantaneous. (Apply)

10. Illustrate the operation of error free sliding window flow control algorithm, with flow diagrams. (Understand)

Course Outcome 6 (CO6)

1. Explain the principle of 1-persistent CSMA/CD. (Understand)
2. What is the minimum size of a frame in 10-Mbps and 1-Gbps Ethernet? (Remember)
3. State the drawback of bus configuration of Ethernet. (Remember)
4. Explain whether a switched gigabit Ethernet requires CSMA/CD. (Understand)
5. Suppose nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 245 bit times. Suppose A and B send Ethernet frames at the same time, the frames collide, and then A and B choose different values of *K* in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at $t = 0$ bit times. They both detect collisions at $t = 245$ bit times. Suppose $KA = 0$ and $KB = 1$. At what time does B schedule its retransmission? At what time does A begin transmission? At what time does A's signal reach B? Does B refrain from transmitting at its scheduled time? (Apply)
6. Suppose nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 325 bit times. Suppose CSMA/CD and Ethernet packets are used for this broadcast channel. Suppose node A begins transmitting a frame and, before it finishes, node B begins transmitting a frame. Can A finish transmitting before it detects that B has transmitted? Why or why not? If the answer is yes, then A incorrectly believes that its frame was successfully transmitted without a collision. *Hint*: Suppose at time $t = 0$ bits, A begins transmitting a frame. In the worst case, A transmits a minimum-sized frame of 512 + 64 bit times. So A would finish transmitting the frame at $t = 512 + 64$ bit times. Thus, the answer is no, if B's signal reaches A before bit time $t = 512 + 64$ bits. In the worst case, when does B's signal reach A? (Analyze)

7. Let A and B be two stations attempting to transmit on an Ethernet. Each has a steady queue of frames ready to send; A's frames will be numbered A1, A2, and so on, and B's similarly. Let T = 51.2 μs be the exponential back off base unit. Suppose A and B simultaneously attempt to send frame 1,collide, and happen to choose back off times of $0\times T$ and $1\times T$,respectively, meaning A wins the race and transmits A1 while B waits. At the end of this transmission, B will attempt to retransmit B1 while A will attempt to transmit A2. These first attempts will collide, but now A backs off for either $0\times T$ or $1\times T$, while B backs off for time equal to one of $0\times T, \ldots, 3\times T$.
   (a) Give the probability that A wins this second back off race immediately after this first collision; that is, A's first choice of back off time $k \times 51.2$ is less than B's.

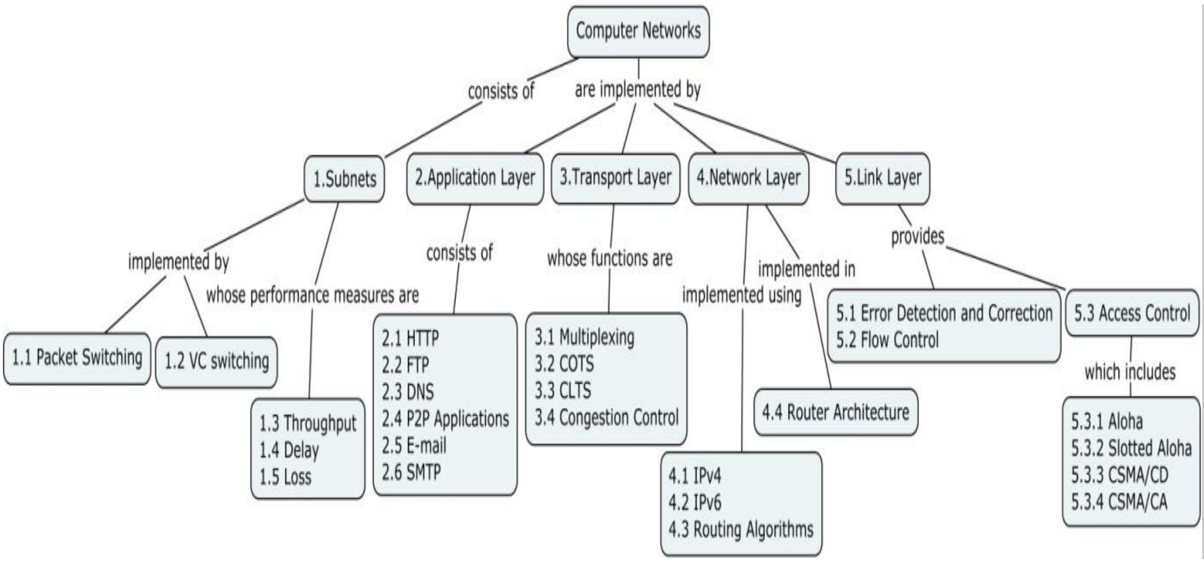
   (b) Suppose A wins this second back off race. A transmits A3, and when it is finished, A and B collide again as A tries to transmit A4 and B tries once more to transmit B1.

(c) Give the probability that A wins this third back off race immediately after the first collision.

(d) Give a reasonable lower bound for the probability that A wins all the remaining back off races. What then happens to the frame B1? (Analyze)

8. Suppose four active nodes—nodes A, B, C and D—are competing for access to a channel using slotted ALOHA. Assume each node has an infinite number of packets to send. Each node attempts to transmit in each slot with probability *p*. The first slot is numbered slot 1, the second slot is numbered slot 2, and so on.
   a) What is the probability that node A succeeds for the first time in slot 5?
   b) What is the probability that some node (A, B, C or D) succeeds in slot 4?
   c) What is the probability that the first success occurs in slot 3?
   d) What is the efficiency of this four-node system? (Apply)

Concept Map



Syllabus

Introduction

Overview of the Internet, Access networks, Packet switching, Circuit switching, Delay, Loss and Throughput in packet networks, Protocol layers and service models

Application layer

Principles of network applications, overview of HTTP, FTP, Electronic mail, Basic operation of SMTP, Overview of DNS and its services, peer to peer applications, Implementation of a simple database in P2P network.

Transport Layer

Transport layer services, overview of transport layer in the Internet, multiplexing and de-multiplexing, principles of reliable and connection oriented transport (TCP)- connectionless transport (UDP), principles of congestion control, TCP congestion control.

Network Layer

Introduction - VC and datagram subnets-Router architecture and processing- The Internet Protocol (IPv4) - addressing and forwarding- Routing algorithms-Introduction to IPv6.

Link Layer

Introduction-Services provided- Flow control-Error detection and correction- Random access- ALOHA-slotted ALOHA – CSMA/CD- Ethernet operation- CSMA/CA – ARP- switched LANs and VLANs.

Text Book

1. James F. Kurose & Keith W. Ross, Computer Networking A Top-down approach, Sixth Edition, Pearson Education ,2013.

Reference Books

1. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, , Mc Graw Hill, 2013.
2. William Stallings, Data and Computer Communications, 8th edition, Pearson Prentice Hall, 2007.
3. Larry L .Peterson and Bruce S. Davie, Computer Networks A Systems Approach, Fifth edition, Morgan Kaufman,2012
4. Andrew S Tanenbaum and David J. Wetherall, Computer Networks, Fifthe Edition, Pearson, 2010

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Subnets**	
1.0	Introduction	1
1.1	Packet Switching	1
1.2	VC switching	1
1.3	Throughput	1
1.4	Delay	1
1.5	Loss	1
2	**Application Layer**	
2.1	HTTP	1
2.2	FTP	1
2.3	DNS	1
2.4	P2P Applications	1
2.5	E-mail	1
2.6	SMTP	1
3	**Transport Layer**	
3.1	Multiplexing	1
3.2	Connection Oriented Transport Service (COTS)	3
3.3	Connection-less Transport Service (CLTS)	2
3.4	Congestion Control	2
4	**Network Layer**	
4.1	IPv4	2
4.2	IPv6	2
4.3	Routing Algorithms	2
4.4	Router Architecture	2
5	**Link Layer**	
5.1	Error Detection and Correction	2
5.2	Flow Control	1
5.3	Access Control	1
5.3.1	Aloha	1
5.3.2	Slotted Aloha	1
5.3.3	CSMA/CD	1
5.3.4	CSMA/CA	1
	Total	**36**

Course Designers:

1. Prof. C.Sridharan cscse@tce.edu
2. Dr. G.S.R.Emil Selvan emil@tce.edu

		Category	L	T	P	Credit
14CS530	**THEORY OF COMPUTATION**	PC	3	0	0	3

Preamble

This course is to introduce students to this fundamental area of computer science which enables students to focus on the study of abstract models of computation. These abstract models allow the students to assess via formal reasoning what could be achieved through computing when they are using it to solve problems in science and engineering. The course exposes students to the computability theory, as well as to the complexity theory. The goal is to allow them to answer fundamental questions about problems, such as whether they can or not be computed, and if they can, how efficiently.

The course introduces basic computation models and their properties, and the necessary mathematical techniques to prove more advanced attributes of these models. The students will be able to express computer science problems as mathematical statements and to formulate proofs.

Prerequisite

- 14C410 : Discrete Mathematics and Combinatorics

Course Outcomes

On the successful completion of the course, students will be able to

Illustrate the capabilities and limitations of the abstract machines including finite automata, pushdown automata, and Turing machines and their associated languages. (CO1)	Understand
Construct finite automata, pushdown automata, Turing machines for the given grammar and vice versa. (CO2)	Apply
Show that a language is not regular / not context-free using pumping lemma. (CO3)	Apply
Outline the characteristics of P, NP and NP Complete problems in the context of Turing machines. (CO4)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M	L	M	M								L
CO2	M	M	M	M								L
CO3	M	M	L	L								L
CO4	M	L										L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	30	30	30	30
Apply	50	50	50	50
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Demonstrate the closure properties of CFLs.
2. State the difference between recursive and recursively enumerable language.
3. Illustrate L_u is recursively enumerable.
4. State the difference between PDA and TM.
5. Explain any two properties of recursive language in detail.

Course Outcome 2 (CO2):

1. Implement a PDA accepting the language L ={ equal number of X's and Y's }by empty store.
2. Construct a PDA that accepts the language generated by grammar with productions S-> aSbb | a.
3. Construct a TM for a language L ={an bn, n≥1}
4. Construct finite automata that accepting {11, 110}*{0}
5. Explain DFA with set of all strings such that that 10th symbol from the right end is '1'.

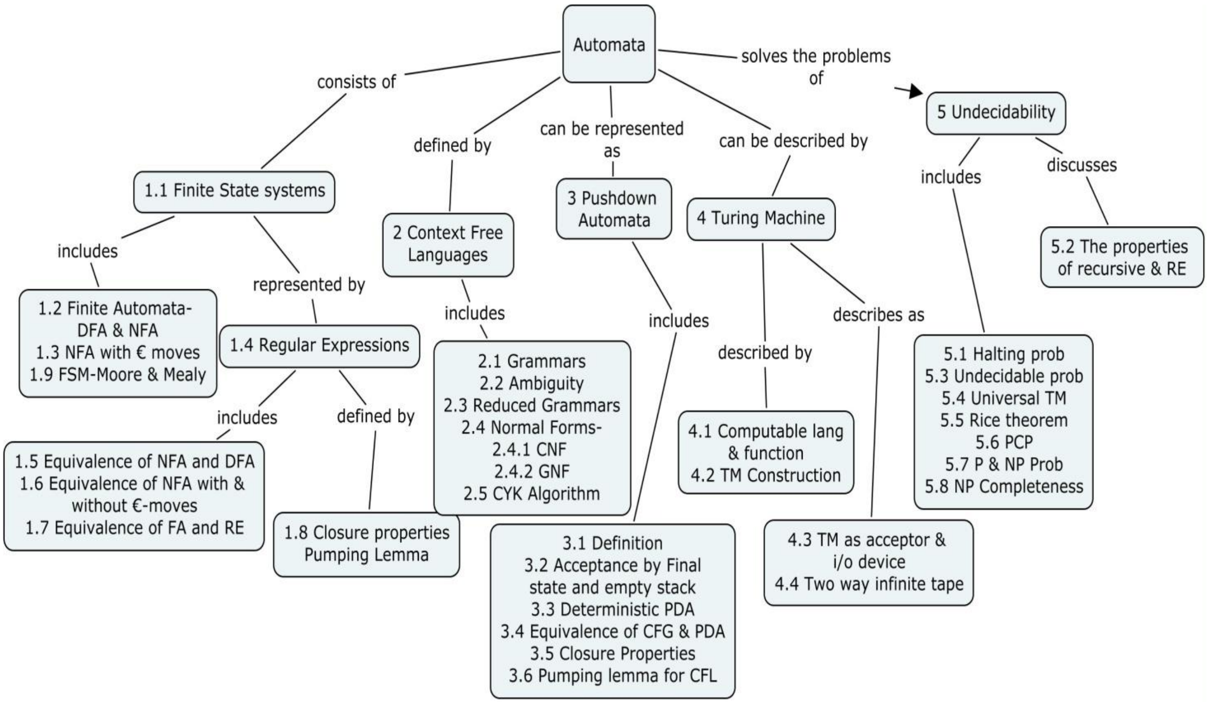
Course Outcome 3 (CO3):

1. Prove that the following languages are not regular using the pumping lemma.
   $L = \{0^n 1^m 0^n | m, n \geq 0\}$
2. Prove that the following languages are not regular using the pumping lemma.
   $L = \{wtw | w, t \in \{0,1\}^+\}$
3. Use the Pumping Lemma for context-free languages to show that the language $L = \left\{a^i b^j c^k \middle| i < j < k\right\}$ is not context-free.

Course Outcome 4 (CO4):

1. Define Class P.
2. Define Class NP.
3. P belongs NP. Why?
4. Describe two classes of decision problem P and NP.
5. Explain that minimum spanning tree is in NP.

Concept Map



Syllabus

Introduction to Finite Automata: Introduction, Finite State systems – Finite Automata, Deterministic finite automata - Non-deterministic finite automata- NFA with € moves – Regular expressions – Equivalence of NFA and DFA – Equivalence of NFA's with and without €-moves – Equivalence of FA and Regular expressions – Closure properties – Pumping lemma for Regular sets – Problems – Finite state machines- moore machine and mealy machines. **Context Free Languages**: Grammars – Derivations and Languages – Relationship between derivation and derivation trees – Ambiguity – Reduced Grammars – Normal forms – Chomsky normal forms – Greiback normal form – CYK algorithm - problems. **Pushdown Automata**: Definitions – Moves – Instantaneous descriptions- Acceptance by final state and empty stack – Deterministic pushdown automata – Equivalence to CFG and Deterministic PDA – Closure properties of CFL - Pumping lemma for CFL – problems. **Turing machines**: Turing machines - Computable languages and functions -Techniques for Turing machine construction – Storage in finite control – multiple tracks – checking of symbols – subroutines - TM as acceptor and i/o device – Two way infinite tape. **Undecidability:** Halting problems – Properties of recursive and Recursively enumerable languages – Universal TM – Decidability – Rice's Theorem – Post's correspondence problem – emptiness and equivalence problems of languages – Time and tape Complexity measure of TM – the classes of P and NP – NP –completeness.

Text Book

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, “Introduction to Automata Theory, Languages and Computations”, second Edition, Pearson Education, 2007.

Reference Books

3. H.R. Lewis and C.H. Papadimitriou, “Elements of the theory of Computation”, Second Edition, Pearson Education, 2003.
4. Thomas A. Sudkamp,” An Introduction to the Theory of Computer Science, Languages and Machines”, Third Edition, Pearson Education, 2007.
5. Raymond Greenlaw an H.James Hoover, “Fundamentals of Theory of Computation, Principles and Practice”, Morgan Kaufmann Publishers, 1998.
6. Micheal Sipser, “Introduction of the Theory and Computation”, Thomson Brokecole, 1997.
7. J. Martin, “Introduction to Languages and the Theory of computation” Third Edition, Tata Mc Graw Hill, 2007.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction to Finite Automata**	
1.1	Introduction, Finite State Systems	1
1.2	Finite Automata - Deterministic finite automata, Non-deterministic finite automata	1
1.3	NFA with € moves	1
1.4	Regular expressions	1
1.5	Equivalence of NFA and DFA	1
1.6	Equivalence of NFA’s with and without €-moves	1
1.7	Equivalence of FA and Regular expressions	1
1.8	Closure properties, Pumping lemma for Regular sets, Problems	1
1.9	Finite state machines- Moore and mealy machines	1
2	**Context Free Languages**	
2.1	Grammars – Derivations and Languages	1
2.2	Relationship between derivation and derivation trees – Ambiguity	1
2.3	Reduced Grammars	1
2.4	Normal forms - CNF. GNF	1
2.4.1	Chomsky normal forms	1
2.4.2	Greiback normal form, Problems	1
2.5	CYK algorithm	1
3	**Pushdown Automata**	
3.1	Definitions, Moves, Instantaneous descriptions	1
3.2	Acceptance by final state and empty store	1
3.3	Deterministic pushdown automata	1
3.4	Equivalence to CFG and Deterministic PDA	1
3.5	Closure properties of CFL	1

3.6	Pumping lemma for CFL – problems	1
4	**Turing Machines**	
4.1	Turing machines - Computable languages and functions	1
4.2	Techniques for Turing machine construction	1
4.2.1	Storage in finite control – multiple tracks – checking of symbols – subroutines	1
4.3	TM as acceptor and i/o device	1
4.4	Two way infinite tape	1
5	**Undecidability**	
5.1	Halting problems	1
5.2	Properties of recursive and Recursively enumerable languages	1
5.3	An undecidable problem that is RE, Undecidable problems about TM	1
5.4	Universal Turing machine	1
5.5	Decidability, Rice's Theorem	1
5.6	Post's correspondence problem (PCP), emptiness and equivalence problems of languages	1
5.7	Time and tape Complexity measure of TM – the classes of P and NP problems	1
5.8	NP –completeness	2
	Total	36

Course Designer

1. Dr.M.K. Kavitha Devi mkkdit@tce.edu

		Category	L	T	P	Credit
14CS540	**COMPUTER ARCHITECTURE**	PC	3	0	0	3

Preamble

This course provides students with a solid understanding of fundamental architectural techniques used to build today's high-performance processors and systems. Course topics include pipelining, superscalar, out of order execution, multithreading, caches, virtual memory, and multiprocessors. Some emphasis will be placed on hardware/software interaction to achieve performance.

Prerequisite

- 14CS230 : Digital Circuits
- 14CS240 : Computer organization and microprocessors

Course Outcomes

On the successful completion of the course, students will be able to

Outline the fundamentals of different instruction set architectures and their relationship to the CPU design. (CO1)	Understand
Apply the arithmetic algorithms to design Arithmetic and Logic Unit (CO2)	Apply
Summarize the ways to take advantage of instruction level parallelism for high performance processor design. (CO3)	Understand
Apply dynamic scheduling methods and show their adaptation to contemporary microprocessor design. (CO4)	Apply
Illustrate the components of memory hierarchy and demonstrate the different cache optimization methods (CO5)	Understand
Design protocols for ensuring cache coherence using the directory based and snooping class of protocols. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M	L										
CO2	S	M	L	L								
CO3	M	L										
CO4	S	M	M	L								
CO5	M	L										
CO6	S	M	M	L								

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	30	20	20

Understand	30	30	40	30
Apply	40	40	40	50
Analyse				
Evaluate				
Create				

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define instruction set architecture (Remember)
2. Show MIPS assembly code that would implement the following High level language code. Use the following register assignments: A is \$t0, B is \$t1, C is \$t2, D is \$t4, R is \$v3.
   A= B+C –D + R (Understand)

Course Outcome 2 (CO2):

1. Assume 185 and 122 are unsigned 8-bit decimal integers. Calculate 185 – 122. Is there Overflow, underflow, or neither. (Apply)
2. Calculate the time necessary to perform a multiply when the 31 adders stacked are Vertically. Assume that an integer is 8 bits wide and an adder takes 4 time units. (Apply)

Course Outcome 3 (CO3)

1. Explain why the MEM and the IF stage of a pipeline can potentially have a structural hazard. What can be done to avoid that hazard? (Understand)
2. What problem can exceptions cause when having a pipelined execution of instruction, and how is the problem typically solved? (Understand)

Course Outcome 4 (CO4)

1. Show what the information table look like for the following code sequence when only the first load has completed and written its result: (Apply)

```
L.D      F6,32(R2)
L.D      F2, 44(R3)
MUL.D    F0,F2,F4
SUB.D    F8,F2, F6
DIV.D    F10,F0.F6
ADD.D    F6,F8,F2
```

2. Examine how data dependencies affect execution of the following code sequence in the basic five-stage pipeline. (Apply)

```
Lw $1, 40 ($6)
add $6, $2, $2
sw $6, 10($3)
```

3. A no pipeline system takes 50 ns to process a task. The same task can be processed in 6 segment pipeline with a clock cycle of 10 ns. Determine the speedup ratio of pipeline for 100 tasks. What is maximum speedup ratio? (Apply)

Course Outcome 5 (CO5)

1. Compare write through Vs Write back cache. (Understand)
2. Why does DMA have priority over CPU when both when both request a memory transfer? (Understand)

Course Outcome 6 (CO6)

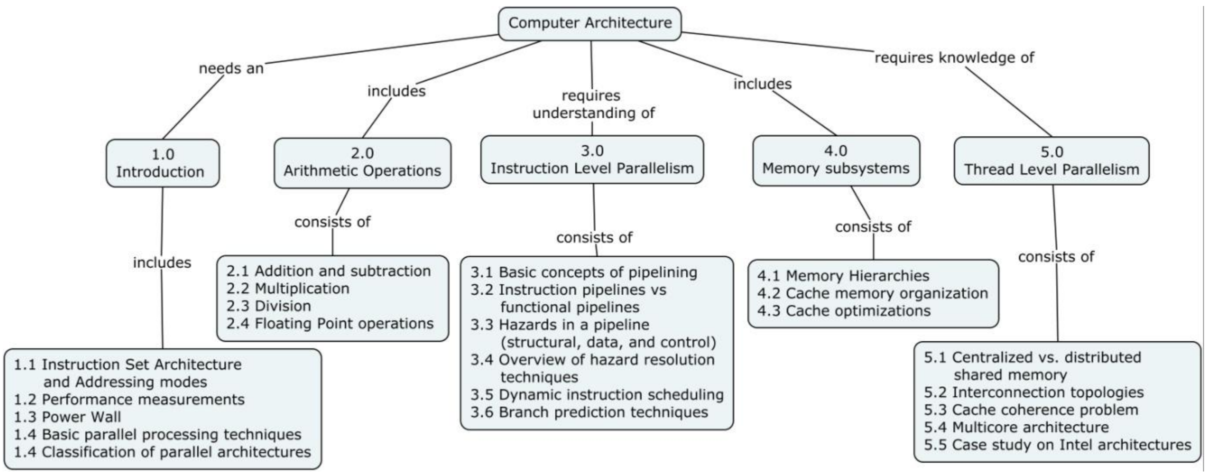
1. Given the following loop sequence.

```
for (i=0; i<8192; i=i+n )
{
ar[i] = c + ar[i];
```

}
with ar being an array of double precision floating point numbers with each element being 8 bytes. No element of the array ar is in the cache before executing this loop. The cache is large enough to hold the entire array ar , and has a cache block size of k = 8 elements (or 64 byt es). Determine the number of cache misses and the cache miss rate for the variable ar in this loop for various loop increment value of n, namely n=1, n=4, and n=16 (Apply)

2. Consider the state of the cache block in “Modified”. Find out the necessary state Transitions and signals generated for the following when MSI protocol is used:
   (i) There is a write made to the block by the same processor
   (ii) There is a write made by the other processor to the same block. (Understand)

Concept Map



Syllabus

Introduction - Instruction Set Architectures and Addressing modes - Performance measurements - Power wall - Basic parallel processing techniques - Classification of parallel architectures. Arithmetic operations - Addition and subtraction - Multiplication - Division - Floating Point operations. Instruction Level Parallelism - Basic concepts of pipelining - Instruction pipelines Vs Functional pipelines - Hazards in a pipeline: structural, data, and control hazards - Overview of hazard resolution techniques - Dynamic instruction scheduling - Branch prediction techniques. Memory Subsystems - Memory hierarchies - Cache memory organization - Cache optimizations. Thread Level Parallelism - Centralized vs. distributed shared memory - Interconnection topologies - Cache coherence problem - Multicore architecture - Case study on Intel architectures.

Text Book

1. Hennessey and Patterson: “Computer Architecture A Quantitative Approach”, 5th Edition, Elsevier,2012

Reference Books

1. David A. Patterson and John L. Hennessey, “Computer organization and design’, Morgan Kauffman / Elsevier, Fifth edition, 2014.
2. V.Carl Hamacher, Zvonko G. Varanesic and Safat G. Zaky, “Computer Organisation“,VI th edition, Mc Graw-Hill Inc, 2012.

Course Contents and Lecture Schedule

Module No.	Topics	No. of Lectures
1	Introduction	
1.1	Instruction Set Architectures and addressing modes	1
1.2	Performance measurements	1
1.3	Power wall	1
1.4	Basic parallel processing techniques	1
1.5	Classification of parallel architectures	1
2.	Arithmetic operations	
2.1	Addition and subtraction	2
2.2	Multiplication	2
2.3	Division	2
2.4	Floating Point operations	2
3.	Instruction level parallelism	
3.1	Basic concepts of pipelining	1
3.2	Instruction pipelines versus functional pipelines	1
3.3	Hazards in a pipeline: structural, data, and control hazards	2
3.4	Overview of hazard resolution techniques	1
3.5	Dynamic instruction scheduling	2
3.6	Branch prediction techniques	2
4.	Memory subsystems	
4.1	Memory hierarchies	1
4.2	Cache memory organization	2
4.3	Cache optimizations	2
5.	Thread level parallelism	
5.1	Centralized vs. distributed shared memory	1
5.2	Interconnection topologies	1
5.4	Cache coherence problem	4
5.5	Multicore architectures	1
5.6	Case study on Intel architectures	2
	Total No. of hours	36

Course Designer

1. Dr. Chitra.P pccse@tce.edu

14CS571	SOFTWARE ENGINEERING: THEORY AND PRACTICE	Category	L	T	P	Credit
		PC	2	0	1	3

Preamble

This subject is to promote the practice of software engineering concepts at a higher level of abstraction, in a more engineering-like fashion. This course focuses on providing hands-on experience in designing and developing software systems. Besides theory sessions, practical sessions are included as part of the study. Consequently students undertake a group project, working through a number of stages of the development of medium size software.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Explain various software development models and processes (CO1)	Understand
Estimate a given project based on Lines of code, Function points and user stories (CO2)	Apply
Explain the process of managing the software quality, change and risk. (CO3)	Understand
Construct Data flow and UML diagrams for a given software requirement specification. (CO4)	Apply
Build a project report as a team which contains the requirement specification, plan, schedule and design documents (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L	L	L		L							M
CO2	M	M	L	L	L						S	M
CO3	M	M	L		L						S	M
CO4	M	M	S	L	S	L	L	L	S	S	S	S
CO5	S	S	S	S	S	S	S	S	S	S	S	S

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3(Practical)	Theory
Remember	20	20	-	20
Understand	30	30	-	30
Apply	50	50	100(Team work)	50
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is the need for engineering software? (Remember)
2. Why should we embrace changing requirements? (Remember)
3. Explain Scrum process in detail. (Understand)

Course Outcome 2 (CO2):

1. What are the causes for variations in project estimates? (Remember)
2. Describe the difference between process and project metrics. (Understand)
3. A system has the following information domain characteristics:

Information Domain Characteristics	Optimistic values	Most likely values	Pessimistic values
Number of external inputs	30	35	46
Number of external outputs	12	15	19
Number of external inquiries	13	17	20
Number of internal files	19	25	30
Number of external interface files	12	16	19

Assume all adjustment factor values are complex (except on-line data entry which takes the values as moderate) and 17 algorithms have been counted. The organizational average productivity for systems of this type is 12.5 FP/pm, and burdened labor rate is 7000$ per month. Estimate the Function Point value, Feature Point value, effort and cost required to build the software.

(Apply)

Course Outcome 3 (CO3):

1. List various software Quality metrics. (Remember)
2. Discuss about Software reviews. (Understand)
3. Discuss about risk management in detail. (Understand)

Course Outcome 4 (CO4):

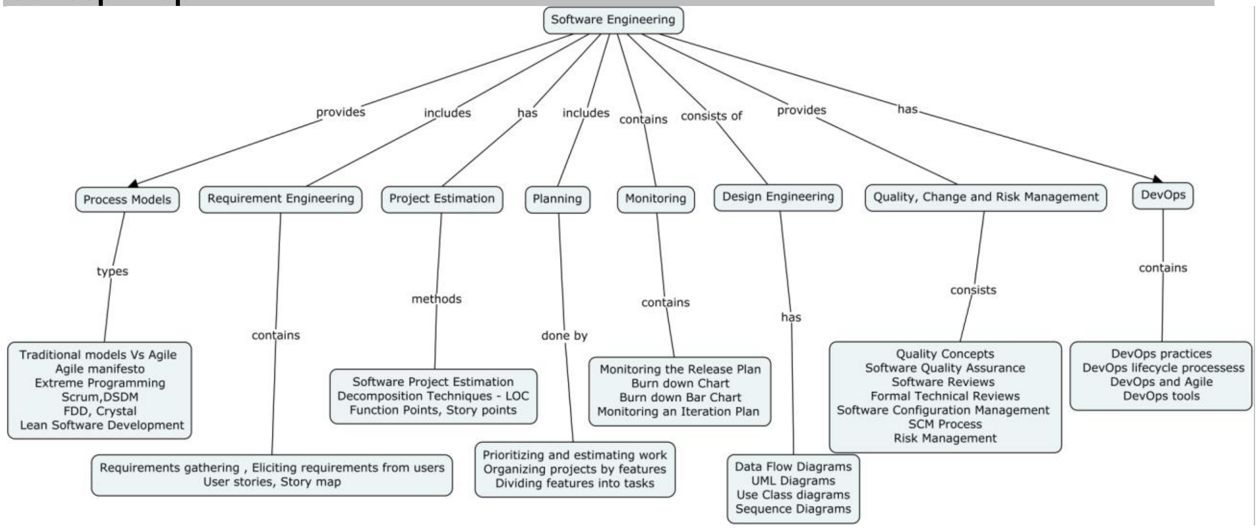
1. List various notations possible in DFD. (Remember)
2. Explain about the various building blocks in UML (Understand)

3. Construct use case diagram for ATM software (Apply)

Course Outcome 5 (CO5):

1. Select software life cycle model suitable for weather monitoring system and justify your answer. (Apply)
2. Develop user stories and story map for weather monitoring system (Apply)
3. Construct a release plan for weather monitoring system (Apply)

Concept Map



Syllabus

Process Models

Traditional models Vs Agile, Agile manifesto, **Agile methodologies-** Extreme Programming, Scrum, DSDM, FDD, Crystal, Lean Software Development.

Requirement Engineering

Requirements gathering , Eliciting requirements from users, user stories, story map

Practical Component

Identifying the requirements from the problem statements and develop user stories.

Project Estimation

Software Project Estimation, Decomposition Techniques - LOC and Function Points, story points.

Practical Component

Estimate a project based on LOC and FP

Planning

Prioritizing and estimating work, organizing projects by features, dividing features into tasks.

Practical Component

Prepare iteration Plan based on story points.

Monitoring

Monitoring the Release Plan - Burn down Chart, Burn down Bar Chart, A Parking Lot Chart. Monitoring an Iteration Plan - The task board, Iteration burn down charts.

Practical Component

Monitor the project using burn down charts.

Design Engineering

Data Flow Diagrams, Introduction to UML Diagrams, Use Case diagrams, Sequence Diagrams.

Practical Component

Modelling Data Flow Diagrams, Modelling UML Use Case Diagrams Modelling Sequence Diagrams.

Quality, Change and Risk Management

Quality Concepts, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Software Configuration Management, SCM Process, Risk Management: Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Migration, Monitoring and Management, RMMM Plan.

Practical Component

Demonstration of Version Control

DevOps

DevOps practices, DevOps lifecycle processess, DevOps and Agile, DevOps tools

Practical Component

Demonstration of DevOps

Text Books

1. Orit Hazzan, Yael Dubinsky, “Agile software engineering”, Springer,2014
2. The Unified Modeling Language Reference Manual, James Rumbaugh, Ivar Jacobson, Grady Booch, 2nd Edition, Addison Wesley,2005.

Reference Books

1. Ian Somerville , "Software Engineering" , 8th Edition,John Wiley and sons,2010.
2. Roger S.Pressman," Software Engineering, A Practitioner’s Approach", 7th Edition, McGraw Hill, 2010.
3. Gene Kim, Jez Humble, Patrick Debois and John willis, "The Devops Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", IT Revolution Press, 2016

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Process Models** (5)	
1.1	Traditional models Vs Agile, Agile manifesto	1
1.2	Agile methodologies**-** Extreme Programming, Scrum, DSDM	2
1.3	FDD, Crystal, Lean Software Development.	2
2	**Requirement Engineering** (2)	
2.1	Requirements gathering , Eliciting requirements from users	1
2.2	User stories, Story map	1
3	**Project Estimation** (3)	
3.1	Software Project Estimation, Decomposition Techniques - LOC	1
3.2	Function Points	1
3.3	Story points	1
4	**Planning** (2)	
4.1	Prioritizing and estimating work, Organizing projects by features	1
4.2	Dividing features into tasks.	1
5	**Monitoring** (2)	
5.1	Monitoring the Release Plan - Burn down Chart, Burn down Bar Chart, A Parking Lot Chart.	1
5.2	Monitoring an Iteration Plan - The task board, Iteration burn down charts.	1
6	**Design Engineering** (3)	
6.1	Data Flow Diagrams	1
6.2	Introduction to UML Diagrams, Use Class diagrams	1
6.3	Sequence Diagrams	1
7	**Quality, Change and Risk Management** (4)	
7.1	Quality Concepts, Software Quality Assurance	1
7.2	Software Reviews , Formal Technical Reviews	1
7.3	Software Configuration Management ,SCM Process	1
7.4	Risk Management	1
8	**DevOps** (3)	
8.1	DevOps practices, DevOps lifecycle processess	1
8.2	DevOps and Agile	1
8.3	DevOps tools	1

Module No.	Topic	No. of Lectures
1	Identifying the requirements from the problem statements and Develop user stories	2
2	Estimate a project based on LOC and FP	2
3	Prepare iteration Plan based on story points.	2

Module No.	Topic	No. of Lectures
4	Monitor the project using burn down charts.	4
5	Modelling Data Flow Diagrams	4
6	Modelling UML Use Case Diagrams	2
7	Modelling Sequence Diagrams	2
8	Demonstration of Version Control	2
9	Demonstration of DevOps tool	4

Course Designers:

1. Mrs. A. Malini amcse@tce.edu
2. Mr. V. Vignaraj Ananth vignaraj@tce.edu

		Category	L	T	P	Credit
14CS580	**DATABASES LAB**	PC	0	0	1	1

Preamble

This course aims at facilitating the student to apply the effective designing of relational database for Real-world applications, perform many operations related to creating, manipulating and maintaining databases using DBMS tools and manipulate data using other languages through ODBC and JDBC.

Prerequisite

14CS440 : Database Management Systems

Course Outcomes

On the successful completion of the course, students will be able to

Build and manipulate relational databases using simple and complex queries in Structured Query Language (SQL). (CO1)	Apply
Develop normalized and denormalized databases for a given application using various constraints like integrity and value constraints. (CO2)	Apply
Construct and make use of database objects such as indices, sequences, synonyms using Structured Query Language (SQL) (CO3)	Apply
Develop objects using PL/SQL and manipulate databases through these objects. (CO4)	Apply
Construct and make use of composite data types using PL/SQL (CO5)	Apply
Develop a complete database application in a high level language using Java Database Connectivity (JDBC) (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L		M				L	L		
CO2	S	M	L		M				M	L		
CO3	S	M			M							
CO4	S	M	L		M				L	L		
CO5	M	L			M							
CO6	S	M	L		M				M	L		

S- Strong; M-Medium; L-Low

List of Experiments

1. Analyze and model a database application.
2. Creation and Modification of Tables without normalization.
3. Creation and Modification of Tables with normalization.
4. Integrity Constraints enforcement.
5. Simple SQL Queries.

6. Complex SQL Queries - I
7. Complex SQL Queries - II
8. Creation and usage of other database objects.
9. Creation of Procedures, Functions and Package with Cursor.
10. Creation of Triggers.
11. Creation of composite data types in PL/SQL.
12. Database application using JDBC.

Course Designers:

1.	Dr. C.Deisy	cdcse@tce.edu
2.	Mrs. A.M.Rajeswari	amrcse@tce.edu

14CS590	NETWORK PROGRAMMING LAB	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

Universal connectivity is realized through Computer Networks. It is important to gain knowledge on the hardware requirements and functioning of Computer Networks. This course provides insight into the working of network protocols and their characteristics.

Course Outcomes

On the successful completion of the course, students will be able to

Implement client server communication using socket programming for various applications. (CO1)	Apply
Configure networking components and installing device drivers and build a Local Area Network. (CO2)	Apply
Perform port scanning and identify IP Address (CO3)	Apply
Implement Remote Method Invocation (CO4)	Apply
Simulate a network topology using NS3(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M	M	S									
CO2	M	M	S									
CO3	M	M	S									
CO4	M	S	M									
CO5	M	M	S									

S- Strong; M-Medium; L-Low

List of Experiments

1. Establishment of a LAN: Preparation of network cables and installation of network device drivers.
2. Write a program to identify your machine's host name and IP address.
3. Write a program to locate the next hop router's IP address and MAC address.
4. Write a program to implement a time server.
5. Write a program to implement ECHO and PING commands.
6. Write a program to illustrate a simple client/server communication.
7. Write a program to obtain local DNS server's host name and IP address.
8. Develop a client that contacts a given DNS server to resolve a given host name.
9. Write a program to implement Remote Method Invocation.
10. Write a program to implement a file transfer using TCP.
11. Write a program to implement a file transfer using UDP.

12. Develop a client server application for chat using Applets.
13. Write a program to find which port is currently used, by scanning the port.
14. Study of NS3/Glomosim network simulators.

Course Designers

1.	Dr. G.S.R. Emil Selvan	emil@tce.edu
2.	Mr. M.P. Ramkumar	ramkumar@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

SIXTH SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

Approved in Board of Studies Meeting on 08.11.14 Approved in 49th Academic Council Meeting on 04.12.14

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY
(For the candidates admitted from 2016-2017)

SIXTH SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS610	Project Management	HSS	3	-	-	3
14CS620	Internet Programming	PC	3	-	-	3
14CS630	Compilers	PC	3	-	-	3
14CSPX0	Program Elective - II	PE	3	-	-	3
14CSGX0	General Elective - I	GE	3	-	-	3
THEORY CUM PRACTICAL						
14CS670	Software Verification and Validation	PC	2	-	2	3
PRACTICAL						
14CS680	Internet Programming Lab	PC	-	-	2	1
14CS690	Engineering by Design Lab	PC	-	-	2	1
Total			**17**	**-**	**6**	**20**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core
PE : Program Elective
GE : General Elective

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture is equivalent to 1 credit
2 Hours Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

SIXTH SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS610	Project Management	3	50	50	100	25	50
2	14CS620	Internet Programming	3	50	50	100	25	50
3	14CS630	Compilers	3	50	50	100	25	50
4	14CSPX0	Program Elective - II	3	50	50	100	25	50
5	14CSGX0	General Elective - I	3	50	50	100	25	50
THEORY CUM PRACTICAL								
7	14CS670	Software Verification and Validation	3	50	50	100	25	50
PRACTICAL								
8	14CS680	Internet Programming Lab	3	50	50	100	25	50
9	14CS690	Engineering by Design Lab	3	50	50	100	25	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

14CS610	PROJECT MANAGEMENT	Category	L	T	P	Credit
		HSS	3	0	0	3

Preamble

The Project Management course discusses activities of planning, organizing, motivating, controlling resources and leadership in theory and practice and the roles and responsibilities of the project manager. It deals with approaches to achieve the project goals and to optimize the allocation of necessary inputs and to integrate them. The course is based on references from Project Management Institute's (PMI®) Project Management Body Of Knowledge (PMBOK® Guide).

Prerequisite

No Prerequisite

Course Outcomes

On the successful completion of the course, students will be able to

Identify a suitable organizational structure for managing projects to develop a product with a given specification (CO1)	Apply
Estimate the cost required to complete a given project by choosing and applying an appropriate cost-estimation method.(CO2)	Apply
Construct a work breakdown structure for a given business cases (CO3)	Apply
Identify the critical path in scheduling a set of project-activities by using the Activity-On-Node method.(CO4)	Apply
Outline the importance and various activities performed for resource management, risk assessment and project closure. (CO5)	Understand
Plan and implement a team-project for developing a product. (CO6*)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.		S						L	M	M	S	
CO2.		L	M					L	M	M	S	
CO3.		S	S						M	M	S	
CO4.		L							M	M	S	
CO5.											M	
CO6.	S	S	S	L	S	S	S	S	S	S	S	S

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests 1	2	3	Terminal Examination
Remember	20	20	20	20
Understand	40	40	80	50
Apply	40	40		30

Analyse				
Evaluate				
Create				

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Suggest a suitable organisational structure for a Central Engineering Systems, Inc.
2. Suggest a leadership style for conveyer belt project

Course Outcome 2 (CO2):

1. Estimate the cost associated with the project using bottom up technique
2. Why is the implementation of projects important to strategic planning and the project manager?
3. What is meant by an integrative approach to project management? Why is this approach important in today's environment?

Course Outcome 3 (CO3):

1. Develop a work breakdown structure for a wedding
2 How does the WBS differ from the project network
3. What is the difference between free slack and total slack?

Course Outcome 4 (CO4):

1. Draw a project network from the following information. What activity(s) is a burst activity? What activity(s) is a merge activity?

ID	Description	Predecessor
A	Survey site	
B	Install drainage	A
C	Install power lines	A
D	Excavate site	B, C
E	Pour foundation	D

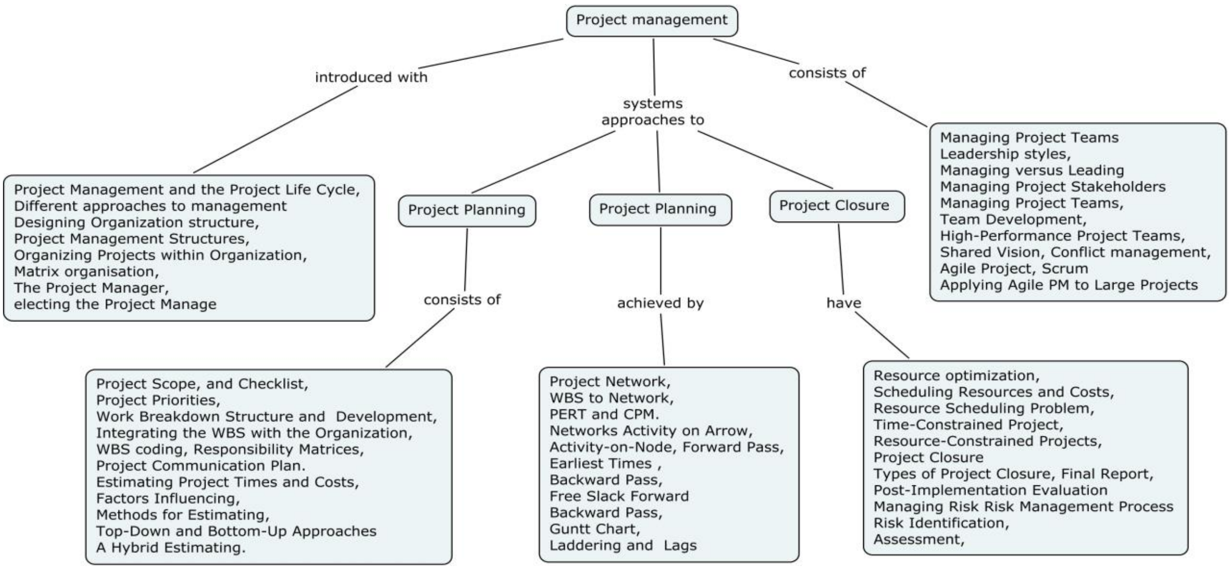
2. Create a customer database for the Modesto league baseball team. Draw a project network Complete the forward and backward pass, compute activity slack, and identify the critical path. How long will this project take? How sensitive is the network schedule? Calculate the free slack and total slack for all noncritical activities.

Course Outcome 5 (CO5):

1 What is the difference between avoiding a risk and accepting a risk?
2 What is the difference between mitigating a risk and contingency planning?
3 Explain the difference between budget reserves and management reserves.
4 How does resource scheduling tie to project priority?
5 How does resource scheduling reduce flexibility in managing projects?
6 What are the advantages of Agile PM? What are the disadvantages of Agile PM?
7 What similarities and differences exist between a traditional project manage and a Scrum master?

*Note: **CO6** will be attained through Mini Projects / Assignments.

Concept Map



Syllabus

Modern Project Management

Project Management and its importance, The Project Life Cycle, Different approaches to management, Integrative Approach. Principles and Steps Designing Organization structure, Project Management Structures, Organizing Projects within the Functional Organization, Matrix organisation, **The Project Manager**, and Project Management, Special Demands Selecting the Project Manager.

Project Initiation

Project Scope, and Checklist, Project Priorities, Work Breakdown Structure and Development, Integrating the WBS with the Organization, WBS coding, Responsibility Matrices, Project Communication Plan. **Estimating Project Times and Costs,** Factors Influencing, Methods for Estimating, Top-Down and Bottom-Up Approaches for Estimating Project Times and Costs, A Hybrid Estimating.

Project Planning

Project Network, WBS to Network, PERT and CPM. Networks Activity on Arrow, Activity-on-Node, Forward Pass, Earliest Times , Backward Pass—Latest Times Determining Slack, Free Slack Forward and Backward Pass, Gantt Chart, Laddering and Lags

Resource optimization, Scheduling Resources and Costs, Resource Scheduling Problem, Time-Constrained Project, Resource-Constrained Projects, **Project Closure** Types of Project Closure, Final Report, Post-Implementation Evaluation. **Managing Risk,** Risk Management Process, Risk Identification, Assessment, and Response Development

Leadership, Leadership styles, Managing versus Leading a Project, Managing Project Stakeholders **Managing Project Teams**, Team Development, High-Performance Project Teams, Shared Vision, Conflict management, **Agile Project**, Scrum Meetings, Applying Agile PM to Large Projects

Text Book

1. Erik W. Larson, Clifford F. Gray, “Project Management The Managerial Process”, McGraw-Hill/Irwin, Fifth Edition, 2011.

Reference Books

1. Jack R. Meredith, Samuel J. Mantel, Jr., "Project management A Managerial Approach"., John Wiley & Sons, Inc. Seventh Edition,2009
2. Harold kerzner , "Project Management A systems approach to Planning, scheduling, And controlling", Tenth edition, John Wiley & Sons, Inc.2009
3. Harold kerzner , "Project management best practices achieving global excellence", Second edition, John Wiley & Sons, Inc.2010
4. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition, Project Management Institute.
5. Harold Koontz, Heinz weihrich "Essentials of Management", Tata McGraw-Hill Education, 2006 - Management

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Modern Project Management**	
1.1	Project Management and its importance	1
1.2	The Project Life Cycle	1
1.3	Different approaches to management, Integrative Approach.	1
1.4	Principles and Steps Designing Organization structure, Project Management Structures,	2
1.5	Organizing Projects within the Functional Organization, Matrix organisation	1
1.6	The Project Manager, and Project Management, Special Demands Selecting the Project Manager	1
2	**Project Initiation**	
2.1	Project Scope, and Checklist, Project Priorities,	1
2.2	Work Breakdown Structure and Development, Integrating the WBS with the Organization, WBS coding.	2
2.3	Responsibility Matrices, Project Communication Plan.	1
2.4	Estimating Project Times and Costs, Factors Influencing, Methods for Estimating,	2
2.5	Top-Down and Bottom-Up Approaches for Estimating Project Times and Costs, A Hybrid Estimating.	2
3.1	**Project Planning**	
3.2	Project Network, WBS to Network	2
3.3	PERT and CPM. Networks Activity on Arrow, Activity-on-Node	2
3.4	Forward Pass, Earliest Times , **B**ackward Pass—Latest Times	2
3.5	Determining Slack, Free Slack Forward and Backward Pass,	2
3.6	Gantt Chart, Laddering and Lags	2
4	**Resource optimization**,	
4.1	Scheduling Resources and Costs, Resource Scheduling Problem,	2
4.2	Time-Constrained Project, Resource-Constrained Projects,	1
4.3	Project Closure Types of Project Closure, Final Report, Post-Implementation Evaluation	1

Module No.	Topic	No. of Lectures
4.4	Managing Risk, Risk Management Process Risk Identification, Assessment, Response Development,	1
5	**Team Management and Agile Management**	
5.1	Leadership styles, Managing versus Leading a Project, Managing Project Stakeholders	1
5.2	Managing Project Teams, Team Development, High-Performance Project Teams, Shared Vision	2
5.3	Conflict management.	1
5.4	Agile Project management, Scrum Meetings, Applying Agile PM to Large Projects	2
	Total	36

Course Designers:

1. Mr. N.Shivakumar, Shiva@tce.edu
2. Mr. V.Vignaraj Ananth Vignaraj@tce.edu

		Category	L	T	P	Credit
14CS620	**INTERNET PROGRAMMING**	PC	3	0	0	3

Preamble

This course is offered in the fifth semester for the students of the computer science and engineering. Students will learn how to represent structure and how to transport data using XML and XML related technologies. Students gain understanding of how the Internet application works and develop programming skills. This course will establish a professional, client-based attitude towards web-design. Students will communicate effectively using today's technologies.

Course Outcomes

On the successful completion of the course, students will be able to

Design web pages using Hyper Text Markup Language (HTML), Dynamic Hyper Text Markup Language (DHTML) (CO1)	Apply
Develop interactive web applications using client side scripting languages (CO2)	Apply
Develop three tier applications using Hypertext Preprocessor (PHP), Java Server Pages (JSP) and Servlets (CO3)	Apply
Construct interoperable web applications using Extensible Markup Language and its related technologies.(CO4)	Apply
Develop and deploy web services to build the server-side components in web applications.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	S		S							
CO2	S	M	S		S							
CO3	S	M	S		S							
CO4	S	M	S		S							
CO5	S	M	S		S							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	20	20	20	20
Apply	60	30	60	40
Analyse				

Evaluate				
Create		30		20

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare the advantages of 3 tier architecture over 2 tier architecture?
2. Compare the differences between IIS/Apache in regards to web development.
3. Write a CSS Rule to change color of all elements containing attribute class = “greenmove” to green and shift them down to 25 pixels and right 15 pixels?

Course Outcome 2 (CO2):

1. Write a Java Script to get input from the user in XHTML and convert to lowercase and uppercase?
2. Write a java script to generate a counter with 2 seconds delay
3. Write a java script to generate Fibonacci series.

Course Outcome 3 (CO3)

1. Write a client server application using Servlet to get radius sent by browser and find area of circle
2. Develop three tier application for online registration of a course using PHP .
3. Develop client server application using JSP for calculating factorial of a number which is sent from client.

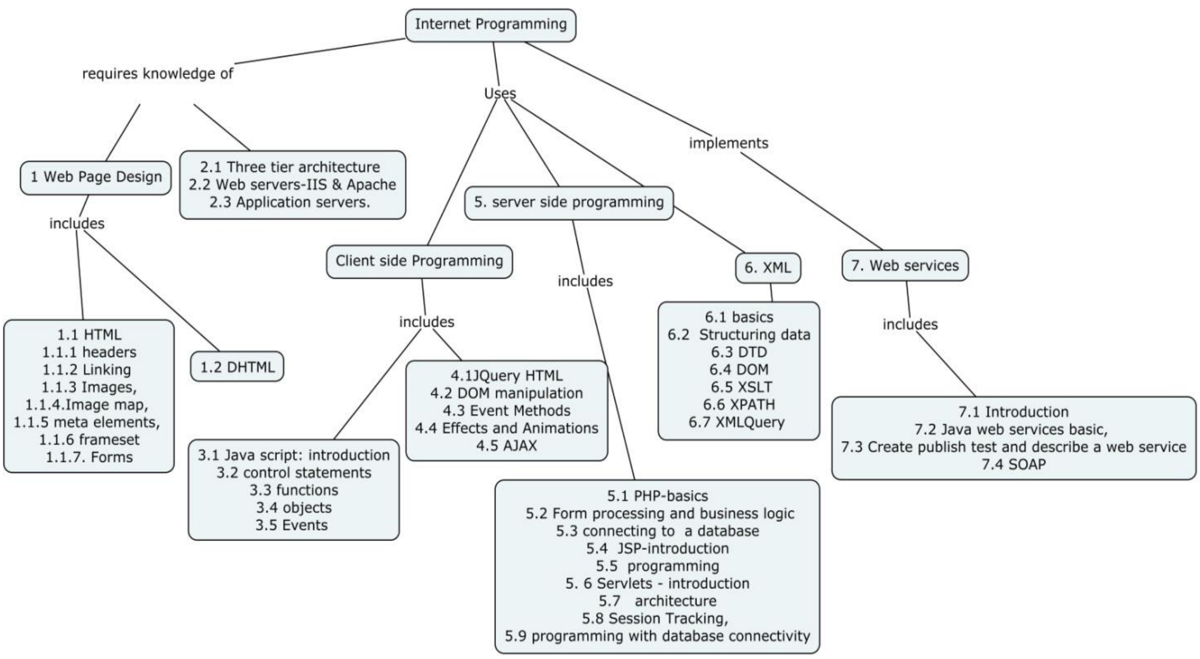
Course Outcome 4 (CO4)

1. Write an XML and DTD for Storing Recipes.
2. Construct the book’s XML document using cascading style sheets.
3. Explain XML Query with an example.

Course Outcome 5 (CO5)

1. Illustrate the principles of publishing in web service.
2. State the purpose of web services.
3. Explain the information stored in the SOAP header.

Concept Map



Syllabus

Web page design: HTML- headers, Linking, Images, Image map, meta elements, frameset, HTML forms, cascading style sheet., DHTML, Two tier and Three tier architecture, J2EE architecture, Web servers-IIS & Apache, Application servers.

Client side programming: Java script: introduction, control statements, functions, objects. Event handling.. JQuery: HTML and DOM manipulation, HTML event methods, Effects and animations, AJAX

Server Side programming: PHP-basics, form processing and business logic, connecting to a database. JSP-introduction, programming, Servlets - introduction, architecture, Session Tracking, Cookies, programming with database connectivity.

XML : XML basics, Structuring data, DTD, DOM, XSLT, XPATH, XML Query

Web services : Introduction, Java web services: basics, Creating, publishing, testing and describing a web service, SOAP.

Text Books

1. Deitel and Deitel, "Internet and World Wide Web How to Program", Prentice Hall of India, Fourth Edition, 2009
2. Paul J.Deitel and Harvey M.Deitel, "AJAX, Rich Internet Applications, and Web Development for Programmers", Pearson Education, First Edition, 2009.

Reference Books

1. Gustavo Alonso, Fabio Casati, Harumi Kuno and Vijay Machiraju, "Web Services" Springer International Edition. First Edition, 2009.
2. Heather Williamson, " XML:The Complete Reference", Tata McGraw Hill,2001
3. www.w3schools.com

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	Web Page Design	
1.1	HTML	
1.2	Headers, Linking,Image maps	1
1.3	Meta elements, Frameset	1
1.4	Forms	1
1.5.	DHTML	1
1.6	Two tier and Three tier architecture	1
1.7	J2EE architecture	1
1.8	Web servers IIS & Apache, Application servers	1
2	Client side programming	
2.1	Java script - Introduction	1
2.2	Control statements	1
2.3	Functions	1
2.4	Objects	1
2.5	Events	1
3	JQuery	
3.1	JQuery HTML, DOM implementation	1

Module No.	Topic	No. of Lectures
3.2	Event Methods	1
3.3	Effects and Animations	1
3.4	AJAX	1
4	PHP	
4.1	PHP basics	1
4.2	Form processing and business logic	1
4.3	Connecting to a database	1
4.4	JSP - Introduction	1
4.5	Programming examples	1
4.6	Servlets - Introduction	1
4.7	Architecture	1
4.8	Session tracking, Cookies	1
4.9	Programming with data base connectivity	1
5	XML	
5.1	Basics	1
5.2	Structuring data	1
5.3	DTD	1
5.4	DOM	1
5.5	XSLT	1
5.6	XPATH	1
5.7	XML query	1
6	Web services	
6.1	Introduction	1
6.2	Java web services basics	1
6.3	Create Publish test web service	1
6.4	SOAP	1

Course Designers

1. Mr. M. Shivakumar, mscse@tce.edu
Mr. T.Manikandan tmcse@tce.edu

		Category	L	T	P	Credit
14CS630	**COMPILERS**	PC	3	0	0	3

Preamble

This course studies programming language translation and compiler design concepts; language recognition, symbol table management, syntax analysis ,code generation and code optimization.

Prerequisite

- 14CS240 : Computer Organization and Microprocessors
- 14CS370 : Object Oriented Programming
- 14CS390 : Assembly Language Programming Lab
- 14CS530 : Theory of Computation

Course Outcomes

On the successful completion of the course, students will be able to

Describe the role of each phase of a compiler with its construction tools. (CO1)	Understand
Develop a Lexical Analyzer for recognizing the tokens of a given language with an understanding of symbol table management and error-handling. (CO2)	Apply
Construct parsers like top-down, bottom-up, operator precedence and SLR with an understanding of Context Free Grammars and syntax analysis (CO3)	Apply
Develop semantic analyzers for type-checking and intermediate code generators to translate the source program into an intermediate code. (CO4)	Apply
Construct code optimizers to optimize the target code generated. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L	L	L									
CO2	S	S	M		S			L	S	L		L
CO3	S	S	M		S			L	S	L		L
CO4	S	S	M		L			L	S	L		L
CO5	S	S	M		L			L	S	L		L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	20	20	20	20
Apply	60	60	60	60
Analyse	--	--	--	--
Evaluate	--	--	--	--
Create	--	--	--	--

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is Token?
2. Give two types of Parsing with example.
3. Define Regular Expression.
4. Define CFG.
5. What is the role of Lexical Analyser?
6. Distinguish between compiler and interpreter.

Course Outcome 2 (CO2):

1. Draw the NFA for recognizing the language (a/b) * abb.
2. Write the regular expression for the language "All strings of digits with even numbers ".
3. Define the term Finite Automata. Distinguish between NFA and DFA.
4. Draw the NFA diagram for the regular expression $011(011)^*$
5. Convert the regular expression of $(a|b)^*abb$ into DFA and draw the minimized transition table.

Course Outcome 3 (CO3)

1. Convert the following operator precedence relations table into precedence graph and construct the precedence functions table.

	(	)	A	,	$
(	<	=	<	<	
)		>		>	>
A		>		>	>
,	<	>	<	>	
$	<		<		

2. Construct the predictive parsing table for the following grammar and check the given grammar is LL(1) grammar or not.

   $S \rightarrow iEtS \mid iEtSeS \mid a$
   $E \rightarrow b$

3. Construct the operator precedence relations table for the following grammar and show the parser movements for the given input string i) *id=id ii) id*id=id
   The grammar is $S \rightarrow L=R$, $S \rightarrow R$, $L \rightarrow *R$, $L \rightarrow id$, $R \rightarrow L$

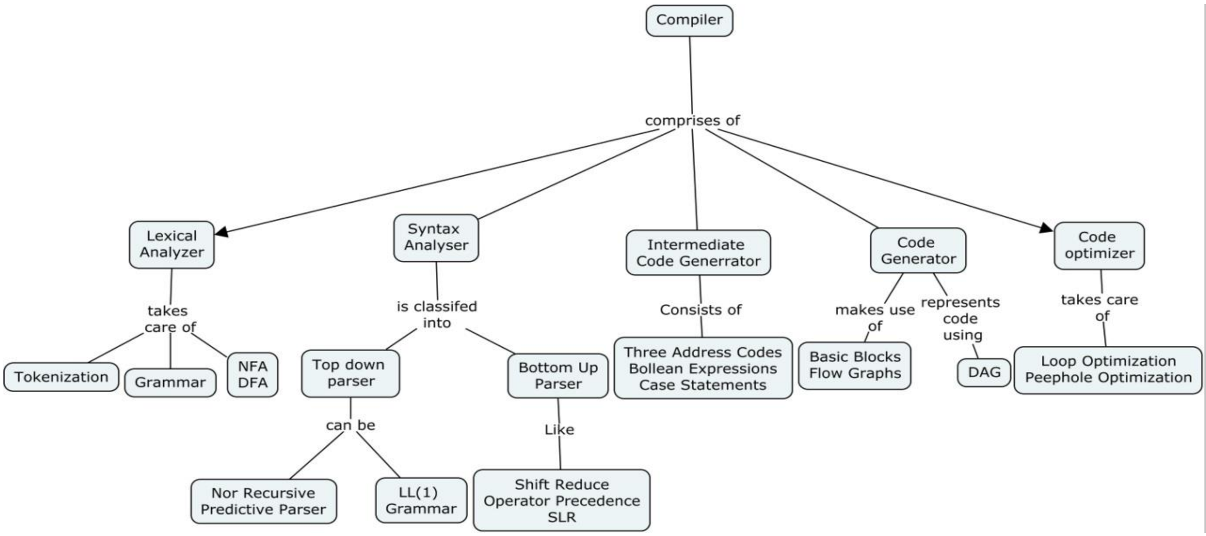
Course Outcome 4 (CO4)

1. Define Back patching? What are the functions are used to implement the Back patching?
2. What is basic blocks? Describe briefly about partition into basic block algorithm.
3. What do you mean by short circuit code representation of Boolean Expression? Give an example.

Course Outcome 5 (CO5)

1. Illustrate benefits of code optimizer.
2. Mention the issues to be considered while applying the techniques for code optimization.
3. Define Intermediate text.
4. Differentiate Triple and Indirect Triple
5. Implement various code optimization techniques in a C program of Quicksort.
6. Write the Principle Sources of Optimization.

Concept Map



Syllabus

Introduction to compiling: Introduction of Compilers-Analysis of the source program-The phases of compiler- The grouping of phases.

Lexical Analysis: The role of the lexical analyzer- Input buffering-Specification of tokens-Recognition of tokens- Regular Expression- Context-free grammars-Writing a grammar-Finite Automata – Conversion of Regular Expression into NFA- Conversion of NFA to minimized DFA.

Syntax Analysis : The role of the parser- Top down parsing -Non-recursive predictive parsing- LL(1) grammar- Bottom-up Parsing- Shift Reduce Parser- Operator precedence parsing-Precedence functions-LR Parser: SLR Parser.

Intermediate Code Generation: Three address Codes- Boolean expressions-Case statements.

Code Generation: basic blocks and flow graphs -The DAG representation of basic blocks-Generating code from DAG's.

Code Optimization: Code improving transformations - Loop optimization -Peephole Optimization.

Text Book

1. Alfred V. Aho, Ravi Sethi, Jeffrey D Ullman – Principles of Compiler Design, Pearson Education, 2nd Edition,2012.

Reference Books

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman- Compilers: Principles, Techniques, and Tools, Pearson Education ,2nd Edition, 2006
2. Dr.O.G.Kakde-Compiler design,4th Edition,Laxmi Publications,2005

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	Introduction to Compiling	
1.1	Introduction of Compilers	1
1.2	Analysis of the source program	1
1.3	Phases of a compiler, Grouping of Phases	1
2.	Lexical Analysis	
2.1	The role of the Lexical Analyser, Input buffering	1
2.2	Specification and recognition of tokens	1
2.3	Regular Expression	1
2.4	Finite Automata	
2.4.1	Conversion of Regular Expression into NFA	2
2.4.2	Conversion of NFA to minimized DFA	2
3.	Syntax Analysis	
3.1	Role of Parsers	1
3.2	Context-Free Grammar, Writing a Grammar	1
3.3	Top-down Parsing	1
3.4	Non-recursive predictive parsing	2
3.5	LL(1) grammar	1
3.6	Bottom-up parsing- Shift Reduce Parser	2
3.7	Operator-precedence parsing	2
3.8	Precedence functions	2
3.9	LR Parser : SLR parser	2
4.	Intermediate Code Generation	
4.1	Three address codes	2
4.2	Boolean expressions	2
4.3	Case statements	1
5.	Code Generation	
5.1	Basic blocks and flow graphs	1
5.2	The DAG representation of basic blocks,	1
5.3	Generating from DAG‘s	1
6	Code Optimization	
6.1	Code improving transformations	1
6.2	Loop optimization	2
6.3	Peephole Optimisation	1
	Total	36

Course Designers:

1. Mr. M.P. Ramkumar ramkumar@tce.edu
2. Mr. R.Chellamani rcmcse@tce.edu

14CS670	SOFTWARE VERIFICATION AND VALIDATION	Category	L	T	P	Credit
		PC	2	0	1	3

Preamble

This course is to encourage students to learn and practice the disciplined approach of software testing. In addition to theory sessions, practical sessions are included as part of the course. Students will learn the concepts of black-box, white-box testing strategies and various testing techniques. Students will apply these testing techniques to small programs and components (functions and classes) and use evaluative techniques such as coverage and mutation testing using various software testing tools.

Prerequisite

- 14CS570 :Software Engineering Theory and Practice

Course Outcomes

On the successful completion of the course, students will be able to

Describe the key techniques and processes involved in software testing. (CO1)	Understand
Construct white-box and black-box test cases using test generation methods like cyclomatic complexity and Finite State Machines. (CO2)	Apply
Determine adequacy for a given test suite using control flow, data flow, and program mutations (CO3)	Apply
Construct prioritized and minimized versions of a test suite using test selection methods. (CO4)	Apply
Develop scripts to automate the testing of a given software using appropriate testing tools. (CO5)	Apply
Construct a test suite to meet the given adequacy criteria involving coverage and mutation score. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L	L	L		M							
CO2	S	S	M	L	S							
CO3	S	S	M	M	S							
CO4	S	S	M	M	S							
CO5	S	S	M	M	S				M	M	M	S
CO6	S	S	M	M	S				M	M	M	S

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3(Practical)	Theory
Remember	20	20	-	20
Understand	20	20	-	20
Apply	60	40	100	50
Analyse	-	-	-	-
Evaluate	-	20	-	10
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate verification with validation. (Remember)
2. List the steps to be performed in Integration Testing Process. (Remember)
3. Explain the various Black Box Testing Techniques. (Understand)

Course Outcome 2 (CO2):

1. Recall about test generation methods from FSM. (Remember)
2. Design a set of test cases for the following program that selects largest of three numbers. (Apply)

```
main()
  {
    float a,b,c;
    scanf ( "%f%f%f,&a,&b,&c);
    If (a>b)
       {  If (a>c)
              printf("%f\n",a);
         else
             printf("%f\n",c);
         } else
             If (c>b)
              printf("%f\n",c);
         else
             printf("%f\n",b);
       }}
```

3. Using Boundary value analysis, design the black-box test suite for a software that computes the square root of an input integer which can assume values in the range of 0 to 5000. (Apply)

Course Outcome 3 (CO3):

1. Recall about Data Flow testing. (Remember)
2. Differentiate dead and live mutant. (Understand)
3. Derive LCSAJ adequate test suite for the above program that selects largest of three numbers. (Apply)

Course Outcome 4 (CO4):

1. Recall about regression test selection problem. (Remember)

2. How to prioritize a set of tests for regression testing? (Understand)
3. Consider a program P has 8 paths and it has been executed against 8 test cases in test suite T. A total of 10 paths are covered by the tests as shown in the following table: (Apply)

Paths								
	P1	P2	P3	P4	P5	P6	P7	P8
Test1	1	0	0	0	0	1	0	0
Test2	0	0	0	1	0	0	0	1
Test3	0	0	1	0	0	1	0	0
Test4	1	1	1	1	0	0	0	0
Test5	1	0	0	0	1	0	0	0
Test6	1	1	1	0	0	0	1	0
Test7	0	0	1	0	0	1	0	0
Test8	0	0	0	0	1	0	0	1

1-> the path is covered by respective test case

0-> the path is not covered by respective test case

Write the procedure CMIMX for test minimization and derive minimal set of test cases from the above test suite T using CMIMX.

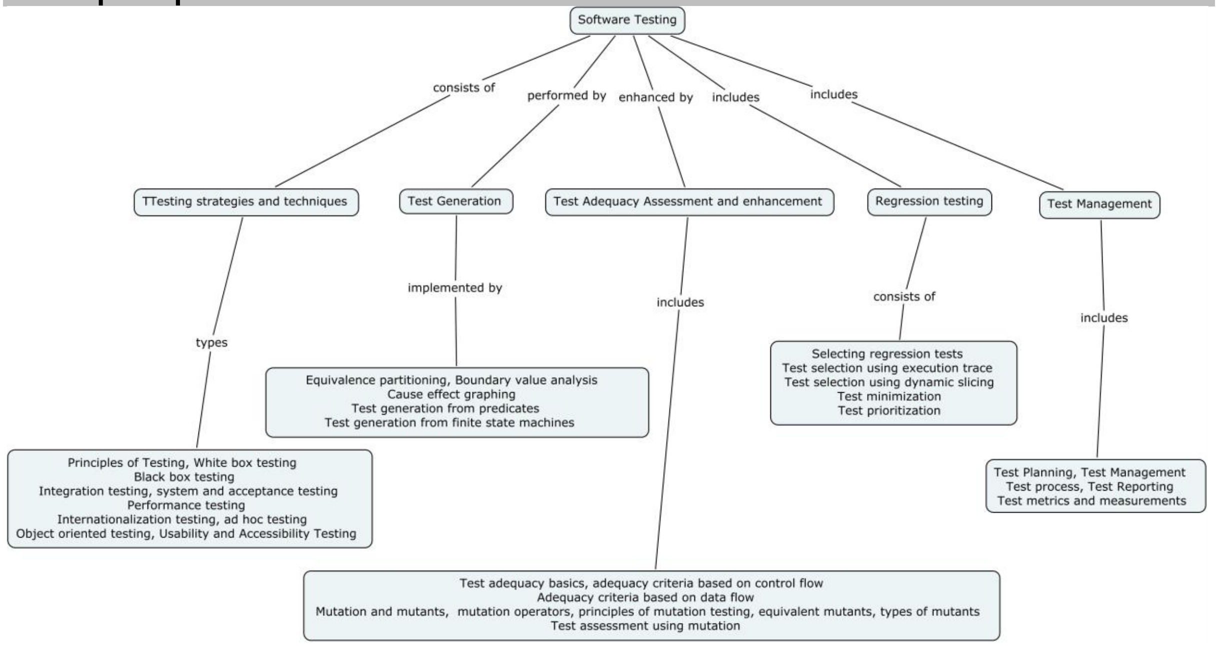
Course Outcome 5 (CO5):

1. Write and execute test scripts using JUNIT framework for the given program. (Apply)
2. Perform GUI and Bitmap checking for given software system using Winrunner Tool. (Apply)
3. Write test scripts to test the functionality of a given software system using Winrunner. (Apply)

Course Outcome 6 (CO6):

1. Estimate the coverage for given test suite using codecover tool. (Apply)
2. Estimate mutation score for given test suite using Muclipse tool. (Apply)
3. Write a test suite to meet given test adequacy criteria using JUNIT. (Apply)

Concept Map



Syllabus

Testing strategies and techniques: Principles of Testing, white box testing, black box testing, integration testing, system and acceptance testing, performance testing, internationalization testing, ad hoc testing, object oriented testing, Usability and Accessibility Testing.

Practical Component:

Test Case generation and execution using Junit Tool

Test Case generation and execution for database-driven software systems using DBUnit Tool.

Performance testing for a given Web Application using JMeter.

Test Generation: Equivalence partitioning, boundary value analysis, cause effect graphing, test generation from predicates, test generation from finite state machines.

Practical Component:

Checking GUI objects and Bitmaps using Win runner tool.

Write Test script Language for given software using Winrunner

Conduct Batch tests using Winrunner

Test Adequacy Assessment and enhancement: Test adequacy basics, adequacy criteria based on control flow, adequacy criteria based on data flow, mutation and mutants, Test assessment using mutation, mutation operators, principles of mutation testing, equivalent mutants, types of mutants.

Practical Component:

Test Case Coverage measurement using Code cover

Mutation testing using Muclipse Tool.

Regression testing and test metrics: Selecting regression tests, test selection using execution trace, test selection using dynamic slicing, test minimization, test prioritization.

Practical Component:

Minimize and prioritise test suite using code coverage and mutation score.

Test Management and Applications: Test Planning, Test Management, Test process, Test Reporting, test metrics and measurements.

Text Books

1. Aditya P. Mathur "Foundations of Software Testing", Second Edition ,Pearson Education, 2014.
2. Srinivasan Desikan, Gopalaswamy Ramesh, "Software testing – principles and practices", First Edition ,Pearson Education, 2009

Reference Book

1. Roger S. Pressman, Software Engineering A Practitioner's Approach, Seventh Edition, Mcgraw Hill International Edition.,2010

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Testing strategies and techniques**	
1.1	Principles of Testing, white box testing	1
1.2	Black box testing	1
1.3	Integration testing, System and acceptance testing,	1
1.4	Performance testing	1
1.5	Internationalization testing, ad hoc testing	1
1.6	Object oriented testing, Usability and Accessibility Testing.	1
2	**Test Generation**	
2.1	Equivalence partitioning, boundary value analysis	1
2.2	Cause effect graphing	1
2.3	Test generation from predicates	1
2.4	Test generation from Finite state machines	2
3	**Test Adequacy Assessment and enhancement**	
3.1	Test adequacy basics, adequacy criteria based on control flow	2
3.2	Adequacy criteria based on data flow	1
3.3	Mutation and mutants, mutation operators, principles of mutation testing, equivalent mutants, types of mutants	1
3.4	Test assessment using mutation	1
4	**Regression testing**	
4.1	Selecting regression tests	1
4.2	test selection using execution trace	1
4.3	Test selection using dynamic slicing	1
4.4	Test minimization	1
4.5	test prioritization	1
5	**Test Management**	
5.1	Test Planning, test Management	1

Module No.	Topic	No. of Lectures
5.2	Test process, Test Reporting	1
5.3	Test metrics and measurements.	1

Module No.	Topic	No. of Lectures
1	Test Case generation and execution using Junit Tool	6
2	Test Case generation and execution for database-driven software systems using DBUnit Tool.	5
3	Performance testing for a given Web Application using JMeter.	3
4	Checking GUI objects and Bitmaps using Win runner tool.	2
5	Write Test script Language for given software using Winrunner	2
6	Conduct Batch tests using Winrunner	1
7	Test Case Coverage measurement using Code cover	2
8	Mutation testing using Muclipse Tool.	2
9	Minimize and prioritise test suite using code coverage and mutation score.	1

Course Designers:

1. Mrs. A. Malini amcse@tce.edu
2. Mr. N. Shivakumar shiva@tce.edu
3. Mr. V. Vignaraj Ananth vignaraj@tce.edu

14CS680	INTERNET PROGRAMMING LAB	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

Students learn how to choose their communication approach by considering platform, dynamically updating the web contents based on the client requirements. They will also learn how to transport data using XML and XML related technologies and protocols and also how to communicate with databases. This course emphasis the working in Application like Ajax, JQuery and also standardization of XML Documents for the purpose of data exchange and integrate the communication mechanism.

Course Outcomes

On the successful completion of the course, students will be able to

Build interactive web pages using Hyper Text Markup Language (HTML),Dynamic Hyper Text Markup Language (DHTML), Cascading Style Sheets (CSS) and Java Script. (CO1)	Apply
Construct three tier applications using Hypertext Preprocessor (PHP) (CO2)	Apply
Construct dynamic web pages using Java Server Pages (JSP) and Servlets (CO 3)	Apply
Install, configure and make use of web servers like Apache, WAMP to host web applications (CO4)	Apply
Utilize Extensible Markup Language (XML) , JQuery and Asynchronous JavaScript and XML (AJAX) in designing web applications (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	M	S		S				S	M		L
CO2.	S	M	S		S				S	M		L
CO3.	S	M	S		S				S	M		L
CO4.	S	M	S		S				S	M		L
CO5	S	M	S		S				S	M		L

S- Strong; M-Medium; L-Low

List of Experiments

1. Create a home page for your project using HTTP and HTML.
2. Create a Webpage to handle Events and Objects using Java Script.
3. Implement Client side form validation.

4. Present a XML document using cascading style sheets (CSS).
5. Implement a program for DOM to Process XML File.
6. Develop a PHP application which connects to the database. (online registration of course)
7. Write a Servlet application which communicates with Browser
8. Develop a Servlet application which connects to the database. (Payroll application)
9. Develop an application to demonstrate the use of AJAX.
10. Develop an application to demonstrate applications of JQuery.

Course Designers

1. Mr. M.Sivakumar mskcse@tce.edu
2. Mr. T.Manikandan tmcse@tce.edu

14CS690	ENGINEERING BY DESIGN LAB	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

This course enables students to gain higher order thinking skills in the core knowledge areas of computer science. This lab represents a confluence of experiments from different special interest groups such as networks, systems, databases, software engineering and multimedia. The lab aims at imparting a hands-on experience in using tools and techniques that will be useful to a student in completing his/her final year project.

Course Outcomes

On the successful completion of the course, students will be able to

Experiment with a wired network to analyze the performance of connection oriented and connectionless transport protocols in a simulated environment. (CO1)	Apply
Develop and Implement parallel algorithms for a given application using MPI and OpenMP. (CO2)	Apply
Make use of any open source video editing tool like Blender, Avidemux to create and modify video files. (CO3)	Apply
Make use of open source visualization tools to present multidimensional data using spotfire, table – lens, parallel co-ordinates, pie chart, bar chart, dendrogram and scatter plot (CO4)	Apply
Design, test and analyze the performance of components of an Operating System like schedulers, I/O drivers and memory managers by modifying the source-code of an open-source operating system, with an understanding of the design trade-offs and hotspots. (CO5)	Analyze
Experiment with concepts of data Integrity and how to manage users in a database system. (CO6)	Analyze
Develop scripts to automate the testing of web applications.(CO7)	Apply

Mapping of Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO3.	S	M	M	L	S			M	S	M		L
CO4.	S	M	M	L	S			M	S	M		L
CO5.			L		S			M	S	M		L
CO6.	S	M	L		S			M	S	M		L

CO7.	S	S	S	M	S			M	S	M		L
CO8.	S	S	M		S			M	S	M		L
CO9.	S	M	M		S			M	S	M		L

L –Low; M – Medium; S – Strong;

List of Experiments

1. Simulation of connection oriented transport protocol and measuring the performance metrics.
2. Simulation of connectionless transport protocol and measuring the performance metrics
3. Develop parallel algorithms for Sieve of Eratosthenes using MPI Library.
4. Design the parallel program for all-pairs shortest path problem using MPI Library.
5. Develop parallel program to perform matrix multiplication using OpenMP constructs.
6. Making of Video or Movie using open source tools
7. Display multidimensional data in different forms using open source visualization toolkit.
8. Design and modification of schedulers for Minix OS
9. Implementation of I/O drivers for Minix OS
10. Design and creation of memory managers for Minix OS
11. Test the data integrity and consistency by using integrity constraints with ONDELETE/NULL options.
12. Create and manage users, groups and their access rights by using DCL commands.
13. Acceptance testing of web application using Selenium.

Course Designers

1. Dr. S. Mercy Shalinie smscse@tce.edu
2. Mr. C. Senthilkumar cskcse@tce.edu
3. Mrs. A.Malini amcse@tce.edu
4. Mrs.A.M.Rajeswari amrcse@tce.edu
5. Dr.P.Chitra pccse@tce.edu
6. Dr.S.Padmavathi spmcse@tce.edu
7. Mrs.S.Sridevi sridevi@tce.edu
8. Dr.K.Sundarakantham kskcse@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

SEVENTH and EIGHTH SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2016-17 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

Approved in Board of Studies Meeting on 08.11.14 Approved in 49th Academic Council Meeting on 04.12.14

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

COURSES OF STUDY

(For the candidates admitted from 2016-2017)

SEVENTH SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CS710	Accounting and Finance	HSS	3	-	-	3
14CS720	Artificial Intelligence	PC	3	-	-	3
14CS730	Distributed Computing	PC	3	-	-	3
14CSPX0	Program Elective - III	PE	3	-	-	3
14CSPX0	Program Elective - IV	PE	3	-	-	3
14CSGX0	General Elective - II	GE	3	-	-	3
PRACTICAL						
14CS780	Artificial Intelligence Lab	PC	-	-	2	1
SPECIAL COURSES						
14CS7C0	Capstone Course – II	PC	-	-	2	2
Total			**18**	**-**	**4**	**21**

EIGHTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
14CSPX0	Program Elective - V	PE	3	-	-	3
14CSPX0	Program Elective – VI	PE	3	-	-	3
14CSPX0	Program Elective - VII	PE	3	-	-	3
PRACTICAL						
14CS880	Project	PC	-	-	24	12
Total			**9**	**-**	**24**	**21**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
PC : Program Core
PE : Program Elective
GE : General Elective

L : Lecture
T : Tutorial
P : Practical

Note:
1 Hour Lecture is equivalent to 1 credit
2 Hours Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2016-17 onwards)

SEVENTH SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CS710	Accounting and Finance	3	50	50	100	25	50
2	14CS720	Artificial Intelligence	3	50	50	100	25	50
3	14CS730	Distributed Computing	3	50	50	100	25	50
4	14CSPX0	Program Elective - III	3	50	50	100	25	50
5	14CSPX0	Program Elective - IV	3	50	50	100	25	50
6	14CSGX0	General Elective - II	3	50	50	100	25	50
PRACTICAL								
7.	14CS780	Artificial Intelligence Lab	3	50	50	100	25	50
8.	14CS7C0	Capstone Course – II	-	100	-	100	-	50

EIGHTH SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment	Terminal Exam *	Max. Marks	Terminal Exam	Total
THEORY								
1	14CSPX0	Program Elective - V	3	50	50	100	25	50
2	14CSPX0	Program Elective – VI	3	50	50	100	25	50
3	14CSPX0	Program Elective - VII	3	50	50	100	25	50
PRACTICAL								
8	14CS880	Project	-	50	50	100	25	50

* Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

		Category	L	T	P	Credit
14CS710	**ACCOUNTING AND FINANCE**	HSS	3	0	0	3

Preamble

Engineering profession involves lots of decision making. The decisions may range from operation to non-operation. For taking decisions of these kinds, an engineer needs among other data about the organization routine operations and non-routine operations. Accounting is a science which provides all the data by recording, classifying, summarizing and interpreting the various transactions taking place in an organization and thereby helps an engineer in taking vital decisions in an effective manner. Finance is an allied but a separate field relying on accounting and enables engineers in taking useful financial and cost related decisions by providing well defined concepts, tools and techniques.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Explain the basic concepts and process of accounting and finance. (CO1)	Understand
Develop trail balance and financial statements like Trading, Profit and Loss accounts, Balance sheet and Cost sheet. (CO2)	Apply
Demonstrate the concepts and operations of budgetary control (CO3)	Understand
Apply techniques like breakeven analysis and budgeting for an organization. (CO4)	Apply
Select the right sources of finance and mobilize the right quantum of finance and make use of them in most profitable investment avenues. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	S	M	M		M	M		M			S	L
CO3.	M					M					M	
CO4.	S	M	M		S	M		M		L	S	L
CO5.	S	M	M			M					S	L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	30	30	30	30
Apply	50	50	50	50
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the term Accounting.
2. List the concepts of accounting.
3. Recall the methods of depreciation.
4. Name the factors causing depreciation.
5. Write the classification of cost.
6. Define the term capital budgeting.

Course Outcome 2 (CO2):

1. Prepare trading account from the information given below and calculate the net profit. Gross profit.....Rs.10,000; Office and administrative expenses .....Rs.1000; selling and distribution expenses ....Rs.500; Interest on investment received...Rs.500; commission received....Rs.200
2. Compare Trading and profit and loss account.
   Compute depreciation for an asset worth Rs.10,000 and having a scrap value of Rs.2,000 and a life time of 4 years under straight line method.
3. Outline the cost classification based on the nature of cost.
4. Apply the net present value method of evaluating investment decision and say whether the following project could be selected for investment.

Year	Cash inflows in Rs.
0	10,000
1	3,000
2	4,000
3	4,000
4	2,000
5	2,000

Course Outcome 3

1. Construct journal entries for the following business transactions.
   a) X brings in cash Rs.10,000 as capital b)purchases land worth Rs.2000, c)He purchases goods worth Rs.5,000,d)He sells goods for Rs.10,000,e)He incurs travelling expenses of Rs.200.
2. Estimate Gross profit and Net profit and the financial position from the following trial balance extracted from the books of Mr.kumar as on 31.12.2010.

Debit Balances	**Amount in Rs.**	**Credit Balances**	**Amount in RS.**
Buildings	30,000	Capital	40,000
Machinery	31,400	Purchase returns	2,000
Furniture	2,000	Sales	2,80,000
Motor car	16,000	Sundry creditors	9,600
Purchases	1,88,000	Discounts received	1,000
Sales return	1,000	Provision for bad debts	6,00
Sundry debtors	30,000		
General expenses	1,6000		
Cash at bank	9,400		
Rates and taxes	1,200		
Bad debts	4,00		
Insurance premium	8,00		
Discount allowed	1,400		
Opening stock	20,000		
Total	**3,33,200**	**Total**	**3,33,200**

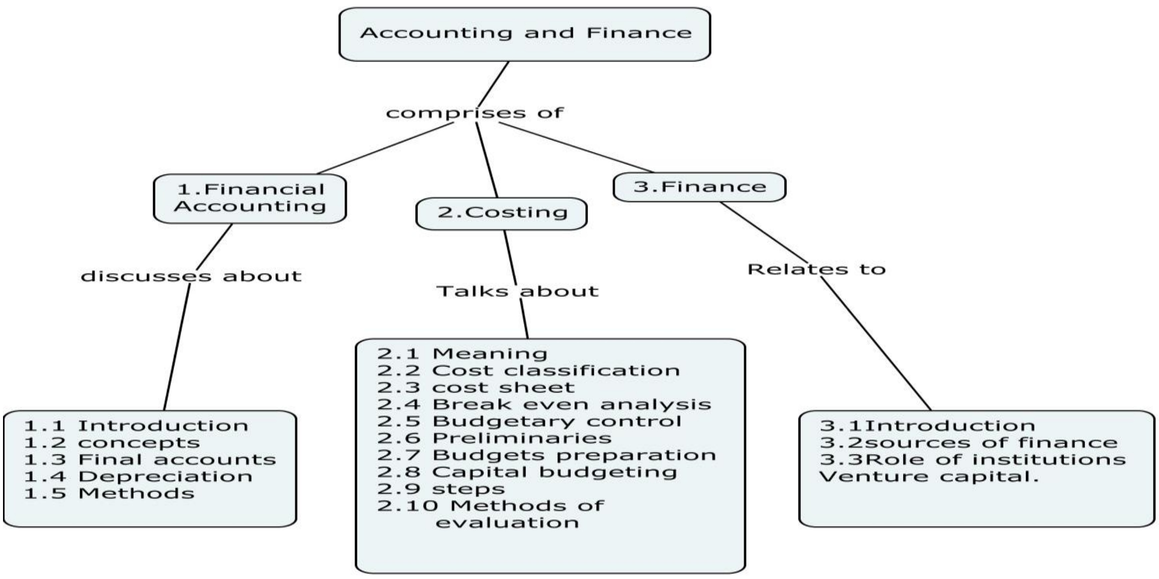
3. Calculate depreciation for a machinery purchased by senthil for Rs.4,00,000 on 1st April 2010.He also adds an additional machinery for Rs.40,000 on 1st April 2011.Depriciation is to be provided at 10% per annum using straight line method. The firm closes its books on 31st March every year.

4. A factory is currently working at 50% capacity and the product cost is Rs.180 per unit as below:
   Material ....Rs.100; Labour.....Rs.30
   Factory overheads....Rs.30 (40% fixed)
   Administration overhead ..Rs.20 (50% fixed)
   The product is sold at Rs.200 per unit and the factory produces 10,000 units at 50% capacity.
   Estimate profit if the factory works to 60% capacity. At 60% working raw material increases by 20% and selling price falls by 20%.

Course Outcome 4

1. From the following information calculate the Breakeven point in terms of units and breakeven point in terms of sales.
   Sales....Rs.10,000; Variable costs Rs.6,000,fixed costs Rs.2000:profit Rs.2,000;No. Of units produced 1,000 units
2. Describe the term ' Breakeven analysis'
3. Calculate the breakeven point and margin of safety from the following information .
   Fixed cost ...Rs.10,000, sales in Rs.25,000,selling price per unit Rs.30; variable cost per unit Rs.10

Concept Map



Syllabus

Accounting: Introduction and Definition-Accounting concepts and conventions-Final Accounts-Preparation of Trading, Profit and Loss Account and Balance Sheet.Depriciation-Meaning-Need and objectives-Basic factors-Methods of providing depreciation.
Cost Accounting: Meaning and Importance-Cost-Elements of cost-Cost classification-Preparation of cost sheet. Break even analysis-Managerial applications. Budget and budgetary control. Meaning- Objectives of budgetary control-Preliminaries for operation of budgetary control-Budgets-Types of budgets and their preparation. Capital budgeting-

Meaning-Importance-steps in capital budgeting-Information needed-Methods of evaluating capital budgeting decisions.
Finance: Introduction-Definition-objectives-functions of finance-sources of finance-Short-term, Medium term, and Long-term-Role of special financial institutions in financing-Venture capital.

Text Books

1. M.C.Shukla,T.S.Grewal,“AdvancedAccounts-Volume-I,2010 Reprint, S. Chand & company Ltd.,2010.
2. Prasanna Chandra, “Financial Management-Theory and practice” seventh Reprint,Tata McGraw-Hill publishing company Limited,2010.

Reference Books

1. A.Ramachandra Aryasri, V.V Ramana Murthy, “Engineering Economics and Financial Accounting, Tata Mc Graw hill, 2010.
2. Dr.V.R.Palanivelu,”Accounting for Management “Third Edition, 2013, University science press New Delhi.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Financial accounting**	
1.1	Introduction and Definition	1
1.2	Accounting concepts and conventions	2
1.3	Final accounts-Preparation of Trading profit and Loss account and Balance sheet.	4
1.4	Depreciation- Meaning, Need and Objectives	2
1.5	Basic factors-Methods of providing depreciation	3
2.	**Cost Accounting**	
2.1	Meaning and Importance	1
2.2	Cost-Elements of cost-Cost classification	2
2.3	Preparation of cost sheet	2
2.4	Break even analysis-Managerial applications	2
2.5	Budget and budgetary control. Meaning- Objectives of budgetary control	1
2.6	Preliminaries for operation of budgetary control	1
2.7	Types of budgets and their preparation	3
2.8	Capital budgeting-Meaning-Importance	1
2.9	Steps in capital budgeting-Information needed	1
2.10	Methods of evaluating capital budgeting decisions. Payback period-Rate of Return-Net present value-Internal Rate of return method	3
3	**Finance**	
3.1	Introduction-Definition-objectives-functions of finance	2
3.2	sources of finance-Short-term, Medium term, and Long-term	2
3.3	Role of special financial institutions in financing-Venture capital.	3
	Total	**36**

Course Designers:

1. S.Dhanasekaran sdmech@tce.edu.
2. P.S.Boopathy Manickam psbmeco@tce.edu

		Category	L	T	P	Credit
14CS720	**ARTIFICIAL INTELLIGENCE**	PC	3	0	0	3

Preamble

This course introduces the basic concepts and techniques of Artificial Intelligence. Artificial intelligence is the sub area of computer science devoted to creating software and hardware to get computers to do things that would be considered intelligent as if people did them. This course will help the students to gain generic problem solving skills that have applicability to a wide range of real-world problems. Students can learn how machines can engage in problem solving, reasoning, learning, and interaction.

Prerequisite

14CS410: Discrete Mathematics and Combinatorics

14CS310: Probability and Statistics

14CS240 : Computer Organization and Microprocessors

Students are expected to have a significant level of programming ability, including some C programming and knowledge of search and data structures, such as balanced binary trees.

Course Outcomes

On the successful completion of the course, students will be able to

Explain the key characteristics of intelligent agents (CO1)	Understand
Solve search problems by applying a suitable search strategy (CO2)	Apply
Find the optimal move for a given game using adversarial search (CO3)	Apply
Construct a plan graph for the given problem like Constraints satisfaction problems and STRIPS problems (CO4)	Apply
Construct knowledge representations using logic to facilitate inference in the given problem domain.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L											
CO2	S	S	M	M					L	L		L
CO3	S	S	S	M					L	L		L
CO4	M	L	M	L					L	L		
CO5	S	S	L	L								

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	30	30	30	30
Apply	40	50	50	50
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List out the goals for a system to be rational.
2. What is the Turing test?
3. What do you mean by utility function?
4. Classify the agents
5. Explain the structure of a simple reflex agent

Course Outcome 2 (CO2):

1. Explain Depth First Search with suitable example.
2. How to measure the problem-solving performance of an algorithm?
3. Describe a state space in which iterative deepening search performs much worse than depth-first search.
4. Implement a genetic algorithm approach to the travelling salesperson problem.
5. How can we avoid ridge and plateau in Hill Climbing?

Course Outcome 3 (CO3)

1. Implement move generators and evaluation functions for the following games: chess, Othello and checkers.
2. Construct a general alpha-beta game-playing agent that uses above implementations
3. Describe state descriptors, move generators and terminal tests for poker game
4. Find an optimal move for two-player game using adversial search technique.

Course Outcome 4 (CO4)

1. Describe the differences and similarities between problem solving and planning
2. What do you mean by action schema?
3. Demonstrate the Graph Plan algorithm.
4. Illustrate the forward state-space search with suitable problem.
5. Construct levels 0, 1 and 2 of the planning graph for the STRIPS problem.

Course Outcome 5 (CO5)

1. List out the steps in knowledge engineering process of a first order logic.
2. Differentiate two quantifiers in the logics.
3. Consider a knowledge base KB that contains he following propositional logic sentences:

$$Q \Rightarrow P$$
$$Q \lor R$$
$$P \Rightarrow \neg Q$$

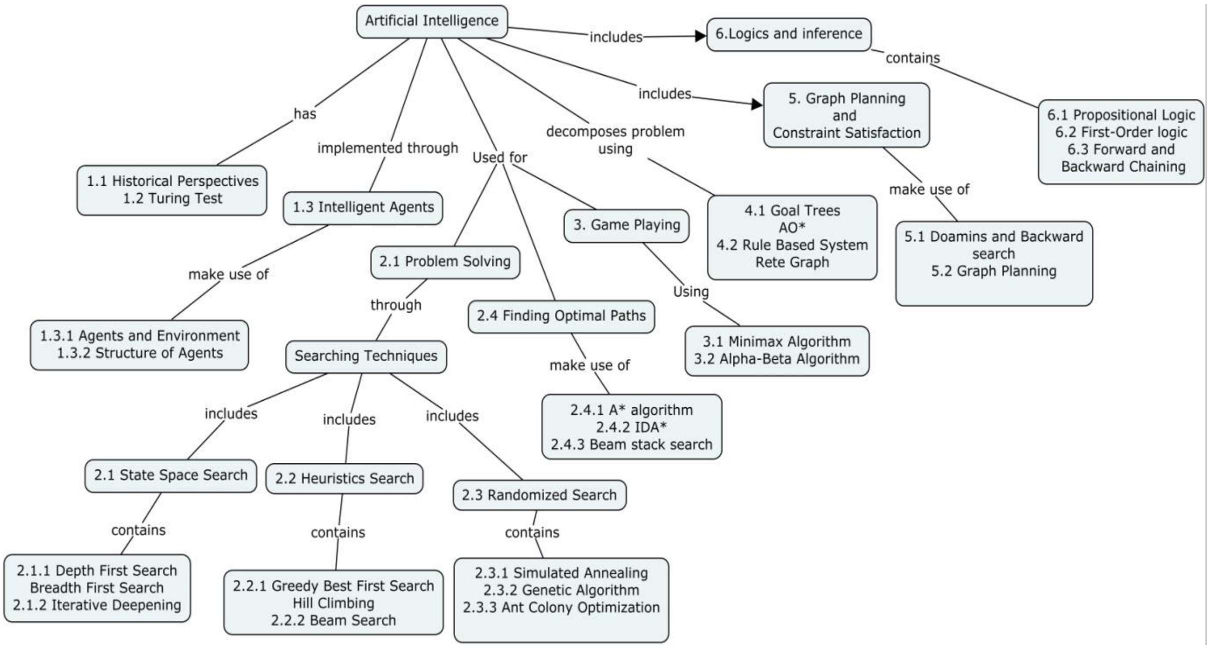
a) Construct a truth able that shows the truth value of each sentence in KB and indicate the models in which the KB is true.

b) Does KB entail Q? Use he definition of entailment to justify your answer.

c) Extend the truth table and use the definition of entailment to justfy your answer.

4. Write down a first-order logic sentence such that every world in which it is true contains Exactly one object in its domain.
5. Suppose that a block-stacking robot with a predicate calculus internal representation needs to look for two blocks (on the table or on top of another block) that are clear (nothing on top) in order to put one on top of the other. Write a first-order predicate calculus representation of the desired blocks.

Concept Map



Syllabus

Introduction: Historical perspective-Turing Test – Intelligent Agents-Problem Solving using Searching: State Space Search-Depth First Search-Breadth First Search-Iterative Deepening Search-Heuristics Search: Greedy Best First Search-Hill Climbing-Beam Search-Randomized Search: Simulated Annealing-Genetic Algorithm-Ant Colony Optimization-Finding Optimal Paths: A* - IDA* - Beam stack search- Game Playing: Minimax Algorithm-Alpha beta Algorithm- Problem decomposition: Goal Trees- AO* - Rule Based Systems- Rete Graph-Planning and Constraint Satisfaction: Domains-Forward and Backward Search-Graph Planning-Constraint Propagation-Logic and Inferences: Propositional Logic-First Order Logic-Forward and Backward chaining

Text Books

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
2. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.

Reference Books

1. Elain Rich and Kevin Knight, “Artificial Intelligence ", Tata McGraw Hill, Third Edition, 2008.

2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Fourth Edition, Pearson Education, 2002.
3. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.
4. Journals- Artificial Intelligence, AI Magazine, IEEE Expert, Machine Learning, Computer Vision Image Processing and Graphics, IEEE Transactions on Neural Networks.
5. NPTEL Lectures

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction**	
1.1	Historical perspectives of Artificial Intellligence	1
1.2	Turing test	1
1.3	Intelligent Agents	1
1.3.1	Agents and Environment	1
1.3.2	Structure of agents	1
2	**Problem solving using Searching**	
2.1	**State Space Search**	
2.1.1	Depth First Search, Breadth First Search	1
2.1.2	Iterative Deepening Search	1
2.2	**Heuristics Search**	
2.2.1	Greedy Best First Search, Hill Climbing	1
2.2.2	Beam Search	1
2.3	**Randomized Search**	
2.3.1	Simulated Annealing	1
2.3.2	Genetic Algorithm	1
2.3.3	Ant Colony Optimization	2
2.4	**Finding Optimal Paths**	
2.4.1	A*	1
2.4.2	IDA*	1
2.4.3	Beam stack search	1
3	**Game Playing**	
3.1	Minimax Algorithm	2
3.2	Alpha beta Algorithm	2
4	**Problem decomposition**	
4.1	Goal Trees, AO*	2
4.2	Rule Based Systems, Rete Graph	2
5	**Graph-Planning and Constraint Satisfaction**	
5.1	Domains-Forward and Backward Search	2
5.2	Graph Planning-Constraint Propagation	4
6	**Logic and Inferences**	
6.1	Propositional Logic	2
6.2	First Order Logic	3
6.3	Forward and Backward chaining	2

Course Designer

1. Dr.S.Padmavathi spmcse@tce.edu

		Category	L	T	P	Credit
14CS730	**DISTRIBUTED COMPUTING**	PC	3	0	0	3

Preamble

This course aims at facilitating students to understand the concept that underlie distributed computing systems along with design and implementation details. Students will learn software components of distributed computing systems in developing applications on various distributed computing platforms or environments

Prerequisite

14CS420 : System Software and Operating System

14CS540 : Computer Architecture

14CS520 : Computer networks

Basic Knowledge in the OS- IPC mechanisms, critical sections , Mutual exclusions and networking and shared memory concepts

Course Outcomes

On the successful completion of the course, students will be able to

Explain the characteristics, models and design issues related to distributed systems (CO1)	Understand
Develop Remote Procedure Call based client-server programs (CO2)	Apply
Identify the requirements of mutual exclusion algorithms (CO3)	Apply
Analyze the message complexity of various deadlock detection and prevention algorithms (CO4)	Analyze
Implement a distributed file system for a given Operating System (CO5)	Apply
Construct a fault tolerant distributed computing system to satisfy the given requirements(CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L											
CO2	S		M		M				M	L		
CO3	M	L	L									
CO4	S	S	L	L								
CO5	S	S	S	M				L	L	L		L
CO6	S	S	S	L				L	L	L		L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	10
Understand	40	20	20	20
Apply	30	40	40	50
Analyse	-	20	20	20
Evaluate	-	-	-	
Create	-	-	-	

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List out the characteristics of Distributed Computing
2. Compare various models of distributed computations
3. Discuss briefly key challenges that one needs to address in the design and development ofdistributed applications.
4. Recall models of Distributed computations
5. Classify Distributed Systems

Course Outcome 2 (CO2):

1. Consider initializing the values of a variable x at the nodes of an n-cube. Process 0 is the leader, broadcasting a value v to initialize the cube. Here n=3 and N = total number of processes = $2^n = 8$
   Find the message complexity and space, time complexity when we broadcast using message passing and shared memory model.
2. Develop an RMI application. server in this example implements a generic compute engine.l The idea is that a client has some CPU-intensive job to do, but does not have the horsepower to do it. So the client encapsulates the task to be done as an object and sends it over to a server to be executed. The compute engine on the server runs the job and returns the results to the client.l The compute engine on the server is totally generic and can execute any kind of task requested by the client
3. Write a simple RMI program that demonstrates the invocation of remote object services. For example, when a client sends a message "Ping", the server responds with "Pong".
4. Illustrate the remote procedure call with a suitable example.

Course Outcome 3 (CO3):

1. State the clock consistency condition
2. Define:Drift and clock skew
3. Why is it difficult to keep a synchronized system of physical clocks in distributed systems?
4. Identify the requirements of a mutual exclusion algorithms

5. Show that in Lamport's algorithm if a site Si is executing the critical section, then Si's request need not be at the top of the request_queue at another site Sj . Is this still true when there are no messages in transit?
6. Consider the following simple method to enforce mutual exclusion: all sites are arranged in a logical ring fashion and a unique token circulates around the ring hopping from a site to another site. When a site needs to executes its CS, it waits for the token, grabs the token, executes the CS, and then dispatches the token to the next site on the ring. If a site does not need the token on its arrival, it immediately dispatches the token to the next site (in zero time).
    1. What is the reponse time when the load is low?
    2. What is the reponse time when the load is heavy?

Assume there are N sites, the message/token delay is T, and the CS execution time is E.

Course Outcome 4 (CO4):

1. Show that, in the AND model, false deadlocks can occur due to deadlock resolution in distributed systems .Can something be done about it or they are bound to happen?
2. Explain global state detection based algorithms
3. Demonstrate the various models of deadlocks
4. Inspect the performance of Diffusion Computation based algorithm and find out its message complexity.

Course Outcome 5 (CO5)

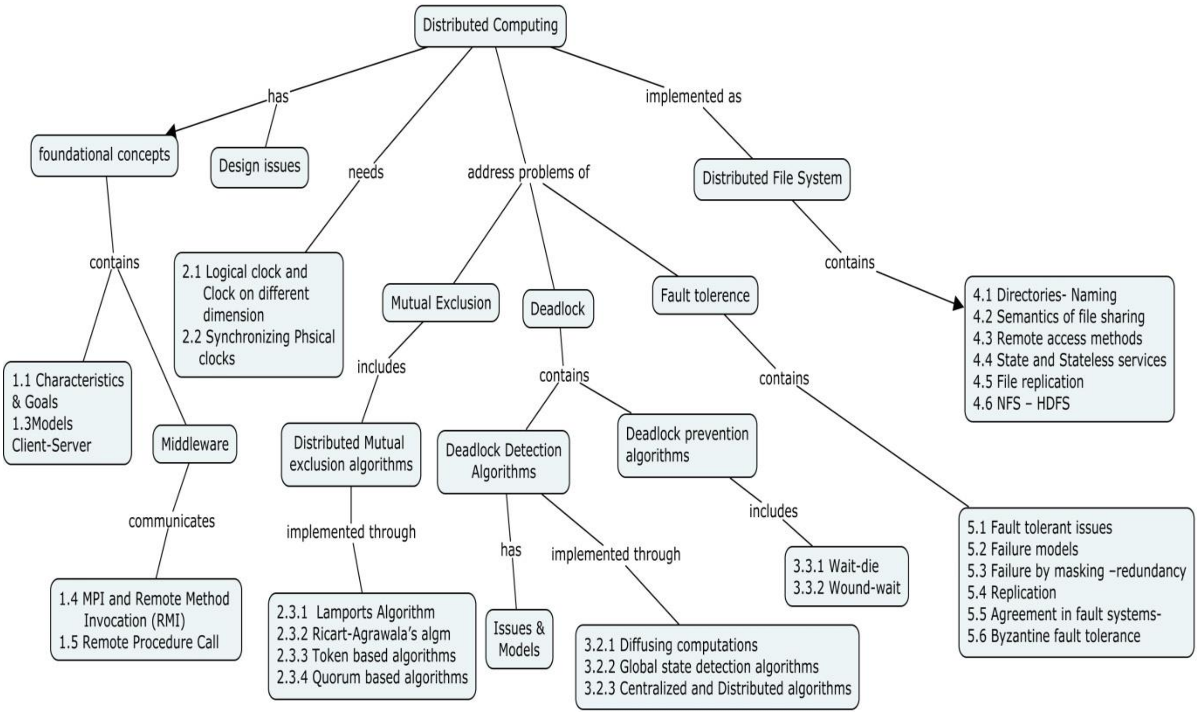
1. Client-side caching allows multiple clients to cache the same file. There are two schemes to validate the contents of a locally-cached file (or invalidate the contents of the same file cached at remote clients.) Those are client-initiated and server-initiated validations. Does the client-initiated validation require a file server to be stateful? Justify your answer. Also show which systems use the client-initiated validation. Does the server-initiated validation require a file server to be stateful? Justify your answer. Also show which system uses the server-initiated validation.
2. Compare caching and remote services of DFS
3. Write about file sharing semantics such as session semantics
4. State one merit for using server-side caching? Which system uses server-side caching?
5. Discuss model architecture of distributed file system and its components
6. Build a Distributed File system that solves complex issues such as uniform name space, location-independent file sharing, client-side caching (with cache consistency), secure authentication

Course Outcome 6 (CO6)

1. How will you achieve fault tolerance?
2. What if server crashes? Can client wait until server comes back up and continue as before?
3. For Scenario: Remote Print Job, when a server crashes and develop different orderings for event

4. List some of types of failures
5. Solve Byzantine failure using agreement in fault tolerant system.

Concept Map



Syllabus

Introduction: characteristics and goals of DS-Design issues- Middlewares- Models-client server- MPI- RMI –RPC- Clock Synchronization: Logical clocks, clocks on different dimensions – Synchronizing physical clocks –- Berkeley algorithm-Cristians Algorithm-Distributed mutual exclusion: Lamports algorithm – Ricart Agrawala's algorithm- token based algorithms-Quorum based algorithms-Deadlock detection in distributed systems: issues and models – algorithms – Diffusing computations –Global state detection algms – centralized and distributed- Deadlock prevention algorithms :wait-die and wound-wait- -Distributed File systems: Directories- naming- semantics of file sharing-Remote access methods – state, stateless services- File replication- NFS – HDFS-Fault tolerance in distributed systems: Fault tolerant issues – failure models — failure by masking –redundancy- Replication agreement in fault systems: Byzantine fault tolerance

Reference Books

1. Ajay D. Kshemkalyani and Mukesh Singhal , "Distributed Computing Principles, Algorithms, and Systems", Cambridge University Press, Fifth Edition,2011.
2. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, "Distributed Systems Concepts and Design", Fifth Edition, Addison Wesley, May 2011
3. Sukumar Ghosh, "Distributed Systems: An Algorithmic approach", 2006 CRC Press
4. M.L.Liu, "Distributed Computing Principles and Applications", Pearson Education, 2004.
5. http://www.cdk5.net/wp/instructors-guide/presentation-points/presentation-pointspresetation-points-chapter-2
6. http://www.slideshare.net/zbigniew.jerzak/clock-synchronization-in-distributed-systems
7. http://www.slideshare.net/sriprasanna/clock-synchronization-distributed-computing

8. http://nptel.ac.in/courses/106106107/
9. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-824-distributed-computer-systems-engineering-spring-2006/

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Basics**	
1.1	Characteristics and Goals of Distributed System	1
1.2	Design issues, Middlewares	1
1.3	Models-client server	1
1.4	MPI- RMI	2
1.5	RPC	2
2.	**Clock Synchronization and Mutual Exclusion**	
2.1	Logical clocks, clocks on different dimensions	1
2.2	Synchronizing physical clocks	
2.2.1	Berkeley algorithm	1
2.2.2	Cristians Algorithm	1
2.3	Distributed mutual exclusion	
2.3.1	Lamports algorithm	1
2.3.2	Ricart-Agrawala's algm	1
.3.3	Token based algorithms	1
2.3.4	Quorum based algorithms	1
3	**Deadlock detection in distributed systems**	
3.1	Issues and Models	1
3.2	Distributed Deadlock detection algorithms	
3.2.1	Diffusing computations	1
3.2.2	Global state detection algorithms	1
3.2.3	Centralized and Distributed algorithms	2
3.3	Deadlock prevention algorithms	
3.3.1	Wait-die	1
3.3.2	Wound-wait	1
4	**Distributed File systems**	
4.1	Directories- Naming	2
4.2	Semantics of file sharing	1
4.3	Remote access methods	1
4.4	State and Stateless services	1
4.5	File replication	1
4.6	NFS – HDFS	2
5	**Fault tolerance in distributed systems**	
5.1	Fault tolerant issues	1
5.2	Failure models	1
5.3	Failure by masking –redundancy	1
5.4	Replication	1
5.5	Agreement in fault systems-	1
5.6	Byzantine fault tolerance	2
	Total	**36**

Course Designer

1. Dr.S.Padmavathi spmcse@tce.edu

14CS780	ARITIFICIAL INTELLIGENCE LAB	Category	L	T	P	Credit
		PC	0	0	1	1

Preamble

The laboratory course will facilitate the Students to apply the concept of artificial intelligence for different problems like eight queens, travelling salesperson problem using machine learning libraries, Python, LISP and PROLOG. These experiments are aimed at imparting a practical exposure to the students to gain generic problem solving skills that have applicability to a wide range of real-world problems. Students can learn how machines can engage in problem solving, reasoning, learning, and interaction.

Prerequisite

14CS720: Artificial Intelligence

Course Outcomes

On the successful completion of the course, students will be able to

Implement breadth first, depth first and best first search technique for problems like 8-puzzle,8-queens,Travelling salesperson and water jug problems etc. (CO1)	Apply
Implement hill climbing, A* algorithm and randomized search techniques for gaming applications.(CO2)	Apply
Develop the solutions for combinatorial problems using intelligent optimization algorithms like Simulated Annealing, Genetic Algorithm, Particle Swarm Optimization. (CO3)	Apply
Construct rule based systems for any application using logic programming language. (CO4)	Apply
Develop solutions for Constraint satisfaction problems using Minimax and Alpha beta pruning algorithms (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	S	M	L								M
CO2.	S	S	M	M	L							M
CO3.	S	S	M	L								M
CO4.	S	S	M	L								M
CO5.	S	S	M	L								M

S- Strong; M-Medium; L-Low

List of Experiments

1. Implement Breadth First Search (for 8 puzzle problem or Water jug problem or any AI search problem)
2. Implement Depth First Search (for 8-queen problem or 8 puzzle problem or Water jug problem or any AI search problem)
3. Solve travelling salesperson problem using Best First Search

4. Implement Hill climbing algorithm
5. Apply any one randomized search technique (Simulated annealing, Genetic Algorithms, Particle swarm optimization) for solving problems like, TSP, Graph coloring, Vertex cover problem, shortest path problems, etc.
6. Write a program to generate the output for A* algorithm.
7. Write a program to show the Tic Tac Toe game for 0 and X
8. Solve the crossword puzzle problem as constraint satisfaction problem
9. Implement anyone Propositional calculus related problem
10. Develop any rule based system for an application of your choice.
11. Generate, view and access decision tree and rules.
12. Implement a k-means clustering algorithm for any given data set.

Course Designer:

1. Dr.S.Padmavathi spmcse@tce.edu

14CS7C0	CAPSTONE COURSE - II

Category	L	T	P	Credit
PC	0	0	2*	2

Preamble

The purpose of this course is to apply the concept of mathematics, science and engineering fundamentals and an engineering specialization to solve complex engineering problems.

Syllabus

Engineering Group1

Databases:
SQL. Normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control.

Theory of Computation:
Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and contex-free languages, pumping lemma. Turing machines and undecidability

Compiler Design:
Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation.

Computer Architecture:
Instruction pipelining, data-path and control unit.

Engineering Group 2

Algorithms :
Algorithm design techniques: greedy, dynamic programming and divide-and-conquer. Graph search, minimum spanning trees, shortest paths.

Operating System:
Processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU scheduling. Memory management and virtual memory. File systems.

Computer Networks:
Concept of layering. LAN technologies (Ethernet). Flow and error control techniques, switching. IPv4/IPv6, routers and routing algorithms (distance vector, link state). TCP/UDP and sockets, congestion control. Application layer protocols (DNS, SMTP, POP, FTP, HTTP). Basics of Wi-Fi. Network security: authentication, basics of public key and private key cryptography, digital signatures and certificates, firewalls.

Assessment Pattern

(Common to B.E./B.Tech Programmes)

Comprehensive Test (30 Marks)

Test 1: Engineering Group 1 (60 Marks) **Duration: 90 Minutes**

Objective Type Questions : 30
Fill up the blanks : 30

Test 2: Engineering Group 2 (60 Marks) **Duration: 90 Minutes**

Objective Type Questions : 30
Fill up the blanks : 30

* - 2 hours/ week is allotted for off-class practical work

Test	Marks Obtained	Converted to
Test1	60 Marks (Max)	15 Marks (Max)
Test 2	60 Marks (Max)	15 Marks (Max)
		30 Marks (Max)

No re-test will be conducted at any circumstances

Complex Engineering Problem Solving (70 Marks):

- Selection of a complex engineering problem (Batch size: 2-4) : 5 Marks
- Literature Survey : 5 Marks
- Problem Formulation : 10 Marks
- Solution Methodology : 15 Marks
- Results and Discussion : 15 Marks
- Technical Report : 10 Marks
- Viva Voce : 10 Marks

Course Designers:

1. Faculty from Engineering Department.

CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAMME

PROGRAMME ELECTIVES

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2016 - 2017 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2008 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

14CSPA0	CRYPTOGRAPHY AND NETWORK SECURITY	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

Cryptography is the science of information and communication security. This course will discuss common security weaknesses, vulnerabilities, attack methods and mitigation approaches in network security. The focus of the course is on Authentication, authorization, confidentiality, data integrity and non-repudiation. Real time network security protocols and system security issues are also addressed here.

Prerequisite

- 14CS520 - Computer Networks

Course Outcomes

On the successful completion of the course, students will be able to

Explain the need for Security Services and Mechanisms to thwart the threats and vulnerabilities of information systems. (CO1)	Understand
Illustrate the techniques that protect and defend information, information systems by ensuring authentication and authorization (CO2)	Understand
Explain the theory of fundamental cryptography, encryption, and decryption algorithms (CO3)	Understand
Apply cryptographic algorithms and Hash algorithms to ensure data secrecy and data integrity. (CO4)	Apply
Develop variants of cryptographic algorithms for the given set of requirements.(CO5)	Analyze
Apply digital signature methods to ensure non-repudiation of data (CO6)	Apply
Illustrate the suitability of security algorithms for real time applications (CO7)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M	L										
CO2.	M	L	L									
CO3.	M	L	L									
CO4.	S	M	L									L
CO5.	S	S	M	M								L
CO6.	S	M	L									L
CO7.	M					M		M	L	L		L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	10
Understand	30	20	20	10
Apply	50	40	40	60
Analyze	0	20	20	20
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare and contrast attack and a threat.
2. Define active attack and its types.
3. State the security mechanisms of OSI standard.

Course Outcome 2 (CO2):

1. Define Entity authentication
2. Define Password Salting
3. State the different Biometrics techniques

Course Outcome 4 (CO3):

1. Differentiate Symmetric and Asymmetric encryption methods.
2. List the participants of Public key cryptosystems.
3. Define cryptanalysis.

Course Outcome 4 (CO4):

1. Eve captures Bob's Hill cipher machine, which uses a 2-by-2 matrix M mod 26. She tries a chosen plaintext attack. She finds that the plaintext ba encrypts to HC and the plaintext zz encrypts to GT. What is the matrix M?
2. Double DES is not used in practice. State reason
3. State reasons for naming the Blowfish algorithm so.
4. Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key and send the result over the channel. Your partner XORs the incoming block with his key and sends it back. You check and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme? If so, what can be done to overcome this flaw?

Course Outcome 5 (CO5):

1. Model an authentication system using RSA algorithm and test with an example.

2. Consider a version of One time pad of Vignere cipher. In this scheme, the key is a stream of random numbers between 0 and 26. For example, if the key is 3 19 5 … then the first letter of plaintext is encrypted with a shift of 3 letters, the second with a shift of 19 letters, the third with a shift of 5 letters, and so on. Encrypt the plaintext 'sendmoney' with the key stream '8 0 1 7 13 15 21 14 11'.

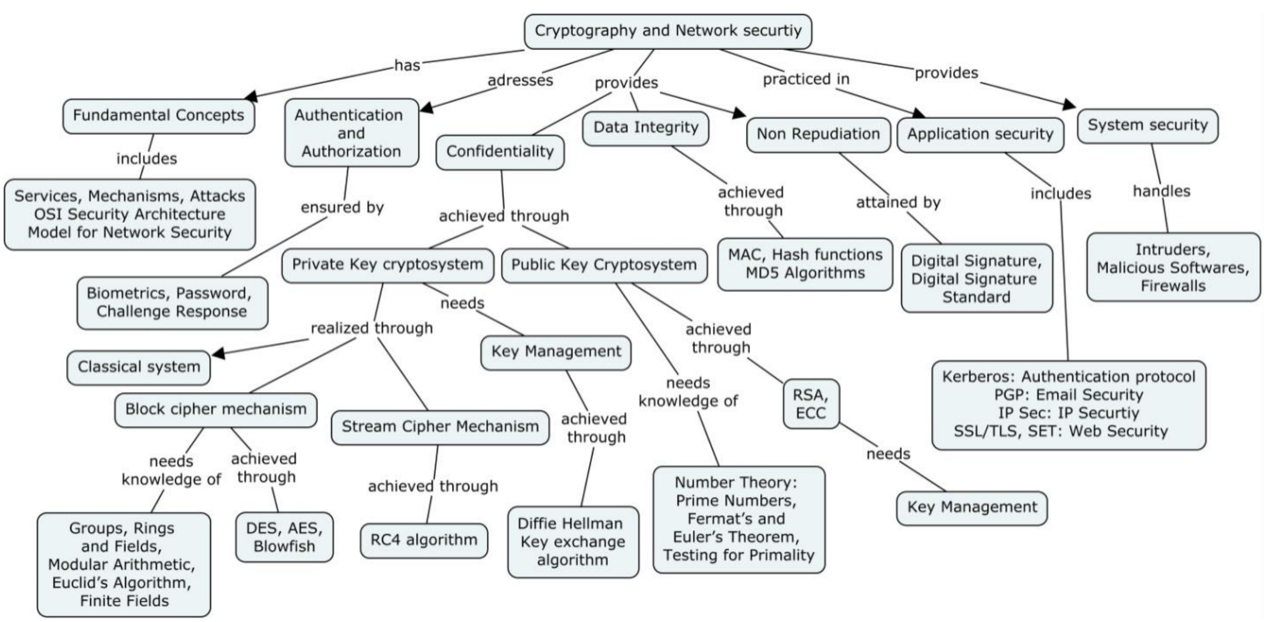
Course Outcome 6 (CO6):

1. Apply RSA algorithm to sign digitally the message PAYRANSOM.
2. Apply PGP authentication and confidentiality services to give message M = 73A56F49257…. K_S=47524635. Given KU_b=5, KR_a=317 n_b=437, KU_a=11, KR_a=35 and n_a=221.
3. Compare and contrast Hash and MAC authentication schemes

Course Outcome 7 (CO7):

1. List the uses of Firewalls.
2. Define any one scheme used in Electronic Mail security.
3. Describe the working of Kerberos 4 authentication protocol with a neat diagram.

Concept Map



Syllabus

Introduction: Services, Mechanisms and Attacks, OSI Security Architecture, Model for Network Security. **Authentication and Authorization:** Biometrics, Password, Challenge Response. **Confidentiality:** General Cipher model, classical encryption techniques, private-key cipher model - block cipher and stream cipher operations, public-key cipher model, attacks on cryptosystems. **Block Cipher Mechanisms:** DES, Block cipher modes of operation. Introduction to Finite Fields: Groups, Rings and Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields, Advanced Encryption Standard, Blowfish. **Stream Cipher Mechanisms:** RC4 Stream Cipher, Key Distribution – Diffie Hellman Key Exchange, Psuedo Random Number Generation. **Public Key Encryption**: Introduction to Number Theory: Prime Numbers, Fermat's and Euler's Theorem, Testing for Primality. Public key ciphers - RSA cryptosystem, Elliptic Curve Cryptography, Key Management. **Data Integrity:** Message Authentication Codes, Hash functions, MD5 Message Digest Algorithm. **Non-Repudiation:** Digital Signature and Digital Signature Standard. **Network Security Practice:** Authentication Application – Kerberos, Electronic Mail Security – PGP, IP Security - IP Security Architecture. Web Security- Secure Socket Layer and Transport layer, Secure Electronic Transaction. System Security: Intruders, Malicious Software, Firewalls.

Reference Books

1. William Stallings, Cryptography and Network Security Principles and Practices, Fourth Edition, Pearson Education, 2008.
2. Behrouz A. Foruzan, Cryptography and Network Security, TataMcGraw Hill, 2007
3. William Stallings, Cryptography and Network Security Principles and Practices, Third Edition, Pearson Education, 2006.

Course Contents and Lecture Schedule

Module No.	Topic	No of Lectures
1	**Fundamental Concepts**	
1.1	Services, Mechanisms and Attacks	1
1.2	OSI Security Architecture	1
1.3	Model for Network Security.	
2	**Authentication and Authorization**	
2.1	Biometrics	1
2.2	Passwords	1
2.3	Challenge Response Schemes	
3	**Confidentiality**	
3.1	General cipher model	1
3.1.1	Classical encryption techniques	2
3.1.2	Private-Key cipher model – block cipher and stream cipher operations	1
3.1.3	Public-Key cipher model	1
3.2	Attacks on Cryptosystems	
3.3	**Block Cipher Mechanisms**	
3.3.1	Data Encryption Standard	2
3.3.1.1	Block cipher modes of operation	1

3.3.2	**Introduction to Finite Fields -** Groups, Rings, Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields	2
3.3.3	Advanced Encryption Standard	2
3.3.4	Blowfish	2
3.4	**Stream Cipher Mechanism**	
3.4.1	RC4 Stream Cipher.	2
3.5	Key Distribution – Diffie Hellman Key Exchange	1
3.6	Pseudo Random Number Generation	
3.6	**Public Key Encryption**	
3.6.1	Introduction to Number Theory: Prime Numbers, Fermat's and Euler's Theorem Testing for Primality	2
3.6.2	RSA	1
3.6.3	Elliptic Curve Cryptography	1
3.6.4	Key Management	1
4	**Message Authentication and Integrity**	
4.1	Message Authentication Codes, Hash functions	1
4.2	MD5 Message Digest Algorithm.	2
5	**Non-Repudiation**	
5.1	Digital Signature and Digital Signature Standard.	1
6	**Network Security Practice**	
6.1	Authentication Application- Kerberos.	1
6.2	Electronic Mail Security- PGP.	
6.3	IP Security- IP Security Architecture.	1
6.4	Web Security- Secure Socket Layer and Transport layer, Secure Electronic Transaction	1
7	**System Security**	
7.1	Intruders	1
7.2	Malicious Software	1
7.3	Firewalls	1
7.4	Password Management.	
	Total	**36**

Course Designer

1. Mrs.M.Suguna mscse@tce.edu

		Category	L	T	P	Credit
14CSPB0	**DATA WAREHOUSING AND MINING**	PE	3	0	0	3

Preamble

This course aims at facilitating the student to understand the concepts of data warehousing and various techniques involved in mining the data.

Prerequisite

14CS440 : Database Management Systems

Course Outcomes

On the successful completion of the course, students will be able to

Build a data warehouse for a given specification and perform various On Line Analytical Processing (OLAP) operations on it. (CO1)	Apply
Prepare a dataset for decision making process by using Integration, Transformation and Discretization methods. (CO2)	Apply
Describe the different data mining techniques and compare data mining systems with database systems (CO3)	Understand
Generate association rules for a given database by applying association rule mining algorithms like Apriori, FP-Growth (CO4)	Apply
Construct a classifier from the given dataset by using classification algorithms like Decision Tree, Naïve bayes and Support Vector Machine(CO5)	Apply
Discover clusters for a given database by applying clustering algorithms like partition based clustering, density based clustering and Conceptual clustering (CO6)	Apply
Describe the concepts of sequence mining, temporal mining, text mining and web mining. (CO7)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L									
CO2	S	M	L	L								L
CO3	M											
CO4	S	M	L	L								L
CO5	S	M	L	L								L
CO6	S	M	L	L								L
CO7	M											

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	10
Understand	40	20	40	30
Apply	40	60	40	60
Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate OLTP and OLAP.
2. Suppose a data warehouse consists of the four dimensions date, spectator, location and game and two measures count and charge , where charge is the fare that a spectator pays when watching a game on a given date. Spectator may be students, adults, or seniors, with each category having its own charge rate. Draw the star schema diagram for the data warehouse.
3. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit. Draw a snowflake schema diagram for the data warehouse.

Course Outcome 2 (CO2)

1. Suppose a group of 12 sales price records has been stored as follows: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215. Partition them into 3 bins by equal width binning.
2. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. Answer the following.
   a) Use smoothing by bin means to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.
   b) What other methods are there for data smoothing?
3. Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000
   a) min-max normalization by setting min = 0 and max = 1
   b) z-score normalization

Course Outcome 3 (CO3)

1. Differentiate DBMS and Data Mining
2. List the issues and challenges of data mining
3. Describe the steps in KDD process

Course Outcome 4 (CO4)

1. For the given database find all the frequent item sets using Apriori method and list all the strong association rules that match the metarule

TID	*Items bought*
100	{*f, a, c, d, g, i, m, p*}
200	{*a, b, c, f, l, m, o*}
300	{*b, f, h, j, o, w*}
400	{*b, c, k, s, p*}
500	{*a, f, c, e, l, p, m, n*}

Minimum Support = 30%

Minimum Confidence = 70%

$\forall$ x € transaction , buys(X,item1) ^ buys(X,item2) $\Rightarrow$ buys(X,item3).

2. Illustrate the significance of candidate set generation step of level wise algorithm.

3. For the given database find all the frequent item sets using FP-growth method. Let the Minimum Support – 30%

TID	Items bought
100	{*f, a, c, d, g, i, m, p*}
200	{*a, b, c, f, l, m, o*}
300	{*b, f, h, j, o, w*}
400	{*b, c, k, s, p*}
500	{*a, f, c, e, l, p, m, n*}

Course Outcome 5 (CO5)

1. For the following Database use ID3 algorithm to construct the decision tree and partition the database based on the classification rules obtained from the decision tree.

Outlook	Temp(F)	Windy	Class
Sunny	medium	True	Play
Sunny	low	False	Play
Sunny	medium	True	No
Sunny	low	True	No
Overcast	high	False	No
Overcast	low	True	Play
Overcast	high	False	Play

Rain	medium	False	Play
Rain	low	False	No
Rain	low	True	Play

2. Describe the importance of pruning in decision tree construction with an example.

3. Compute the value of ‘Flu’ attribute for the following test tuple using Naïve bayes Classification by considering the given database.

X = (Age = Middle-aged, Competition = Yes, Type = Hardware, Profit = ?)

Age	Competition	Type	Profit
Senior	Yes	Software	Down
Senior	No	Software	Down
Senior	No	Hardware	Down
Middle-aged	Yes	Software	Down
Middle-aged	Yes	Hardware	Down
Middle-aged	No	Hardware	Up
Middle-aged	No	Software	Up
Youth	Yes	Software	Up
Youth	No	Hardware	Up

Course Outcome 6 (CO6)

1. Given two objects A1(22,1,42,10) and A2(20,0,36,8) compute the distance by Euclidean measure.

2. Consider ε = 3 units and Minpts = 4. Apply the DBSCAN algorithm to cluster the following points: A(2,2), B(3,1), C(3,4), D(5,3), E(9,8), F(10,7), G(10,10), H(12,8),

   I(3,14), J(10,14), K(11,13), L(12,15), M(14,15).

3. Briefly outline how to compute the *dissimilarity* between objects described by the following types of variables:

   a. Numerical (interval-scaled) variables

   b. Binary variables

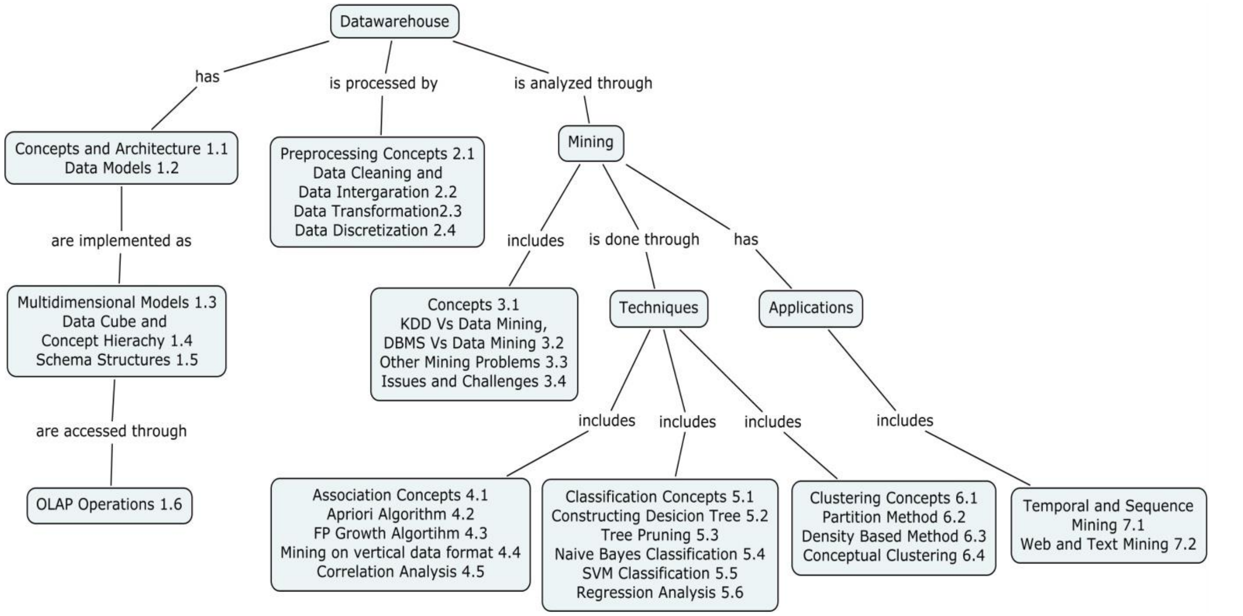
Course Outcome 7 (CO7)

1. Outline the preprocessing steps in text mining

2. Explain different types of web mining techniques.

3. Describe the different levels of temporality in data.

Concept Map



Syllabus

Introduction to Data warehouse – Concepts, Architecture, Data Models, Multidimensional Models - Data Cube, Concept Hierarchy , Schema Structures, OLAP operations **Preprocessing -** Preprocessing Concepts, Data Cleaning, Integration, Transformation and Discretization **Introduction to Data Mining –** Concepts, KDD Vs Data mining, DBMS Vs Data Mining , Other Mining Problems , Issues and Challenges **Association Techniques -** Introduction to Association Rules, Apriori algorithm, FP Growth , Mining on vertical data format, Correlation Analysis **Classification Techniques-** Introduction to Classification **-** Constructing decision tree (ID3 Algorithm), Pruning, Naive Bayes Classification, SVM Classification, Regression Analysis **Clustering Techniques -** Introduction to Clustering , Partitioning Method – K Means algorithm , Density Based Method – DBSCAN method , Conceptual clustering – COBWEB algorithm **Applications of Data Mining -** Temporal Mining, Sequence Mining, Web Mining and Text Mining.

Text Books

1. Arun K.Pujari, “Data Mining Techniques”, Second Edition, Universities Press, 2010.
2. K.P.Soman, Shyam Diwakar, V.Ajay, “Insight into Data Mining – Theory and Practice”, Prentice Hall of India, 2006.

Reference Books

1. Jiawei Han, Micheline Kamper, Jian Pei, “Data Mining: Concepts and Techniques “, Morgan Kaufman, Third Edition, 2011.

2. Ian H. Witten , Eibe Frank , Mark Hall , “Data Mining: Practical Machine Learning Tools And Techniques”, ELSEVIER INDIA PVT . LTD, 2011.

3. M.H Dunham, “Data Mining: Introductory and advanced topics”, Pearson Education, 2006**.**

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction to Data warehouse**	
1.1	Concepts and Architecture	1
1.2	Data Models	1
1.3	Multidimensional Model	
1.4	Data Cube and Concept Hierarchy	1
1.5	Schema Structures	1
1.6	OLAP operations	2
2	**Preprocessing**	
2.1	Preprocessing Concepts	1
2.2	Data Cleaning and Integration	1
2.3	Data Transformation	1
2.4	Data Discretization	2
3	**Introduction to Data Mining**	
3.1	Concepts	1
3.2	KDD Vs Data Mining, DBMS vs Data mining	1

Module No.	Topic	No. of Lectures
3.3	Other Mining Problems	1
3.4	Issues and Challenges	
4	**Association Techniques**	
4.1	Introduction to Association Rules	1
4.2	Apriori algorithm	1
4.3	FP Growth Algorithm	2
4.4	Mining on vertical data format	1
4.5	Correlation Analysis	1
5	**Classification Techniques**	
5.1	Classification Concepts	1
5.2	Constructing decision tree – ID3 algorithm	2
5.3	Tree Pruning	1
5.4	Naive Bayes Classification	1
5.5	SVM Classification	1
5.6	Regression analysis	1
6	**Clustering Techniques**	
6.1	Clustering Concepts	1
6.2	Partitioning Method – K-Means Algorithm	1

Module No.	Topic	No. of Lectures
6.3	Density Based Method – DBSCAN Algorithm	1
6.4	Conceptual clustering – COBWEB Algorithm	2
7	**Applications of Data Mining**	
7.1	Temporal and Sequence Mining	2
7.2	Web and Text Mining	2
	Total	**36**

Course Designer

1. Mrs. B.Subbulakshmi bscse@tce.edu

14CSPC0	**WIRELESS NETWORKS**	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course on Wireless Networks provides an introduction to the basic concepts in wireless networks, architecture and topologies. The objective of this course is to introduce the concepts in wireless Networks and also discuss the main issues in wireless networks such as mobility management , power management and security. At the end of the course, the students should have an understanding of the basic principles of wireless Networks.

Prerequisite

14CS520 : Computer Networks

Course Outcomes

On the successful completion of the course, students will be able to

Explain the propagation mechanisms, generations, wireless transmission techniques, power management, principles and standards of cellular and ad hoc networks (CO1)	Understand
Select the multiple access technique which enables the maximum number of concurrent users for a given specification .(CO2)	Apply
Develop a suitable mechanism to increase the channel utilisation for the given transmission range.(CO3)	Apply
Select the suitable hand off algorithm for the given scenario. (CO4)	Apply
Identify the current location of mobile users using agent advertisements (CO5)	Apply
Make use of different key sizes for enhancement in security for the given scenario(CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	M											
CO3.	S	M	L									L
CO4.	S	M	L									L
CO5.	S	M	L									L
CO6.	S	M	L									L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	50	50	40	40
Apply	20	30	40	40
Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List the different radio propagation mechanisms.
2. List the difference between ad hoc and infrastructure based networks.
3. Give three reasons why it is difficult to detect collisions at the transmitter in wireless networks?

Course Outcome 2 (CO2)

1. Compute the capacity of IS-95 CDMA for the following specifications W=1.5 MHZ, R=9600 bps, Sr=4, Gv=2.5 , Ga=2.75 and Ho=1.67
2. Draw the representation for CDMA/TDD for 5 users.
3. Draw the representation for FDMA/TDD for 6 users.

Course Outcome 3 (CO3)

1. We want to provide a Radio communication service to a city. The total BW available is 25 MHZ, and each user requires 20 KHZ of BW for voicecommunication. If we employ a cellular topology where 25 low power antennas are located and the cluster size is 5. Calculate the following
   a) Number of simultaneous user when one antenna is used to cover the entire city
   b) Number of simultaneous user per cell
   c) Number of simultaneous user per cluster
   d) Number of simultaneous user after cellular topology employed
   e) calculate the ratio when compared to one antenna used.
2. Assume that you have six sector cells in a hexagonal geometry. Assume 60 degree and 120 degree directional antennas are used. Compute Sr in dB for reuse factors of 7,4 and 3. Draw the hexagonal grid corresponding to these cases. Comment on your results.
3. Assume that you have six sector cells in a hexagonal geometry. Assume 60 degree and 120 degree directional antennas are used. Compute Sr in dB for reuse factors of 7,4 and 3. Draw the hexagonal grid corresponding to these cases. Comment on your results.

Course Outcome 4 (CO4)

1. State the different handoff algorithms
2. Define location management**.**

3. A mobile terminal samples signals from four BS as a function of time. The times and signal strengths from the samples are given in Table. Assume the mobile terminal is initially attached to BS_1.The mobile makes handoff decisions by considering the signals from BS's after each sampling time. Show the handoff transitions between BSs for each of the following algorithms as a function of time. Find out the BS selected after each and every 2.5 seconds. If a condition is met for more than one BS, assume the best one (Strongest RSS) is selected.
   I. Received signal strength (RSS)
   II. RSS + Threshold of 1000 x 10^{-12}w
   III. RSS +Hysteresis of 10^{-9} watts

Time(sec)	0	2.5	5	7.5	10	15
BS_1	15 X 10^{-9}	0.15 X 10^{-9}	2 X 10^{-9}	1 X 10^{-9}	90 x10^{-12}	90 x10^{-12}
BS_2	1 X 10^{-9}	0.8X 10^{-9}	0.8X 10^{-9}	0.5 X 10^{-9}	1 X 10^{-9}	80 x10^{-12}
BS_3	205 x10^{-12}	1X 10^{-9}	0.9X 10^{-9}	2.5X 10^{-9}	0.5 X 10^{-9}	0.5X 10^{-9}
BS_4	200X10^{-12}	110X10^{-12}	110X10^{-12}	90 x10^{-12}	0.75X 10^{-9}	0.75X 10^{-9}

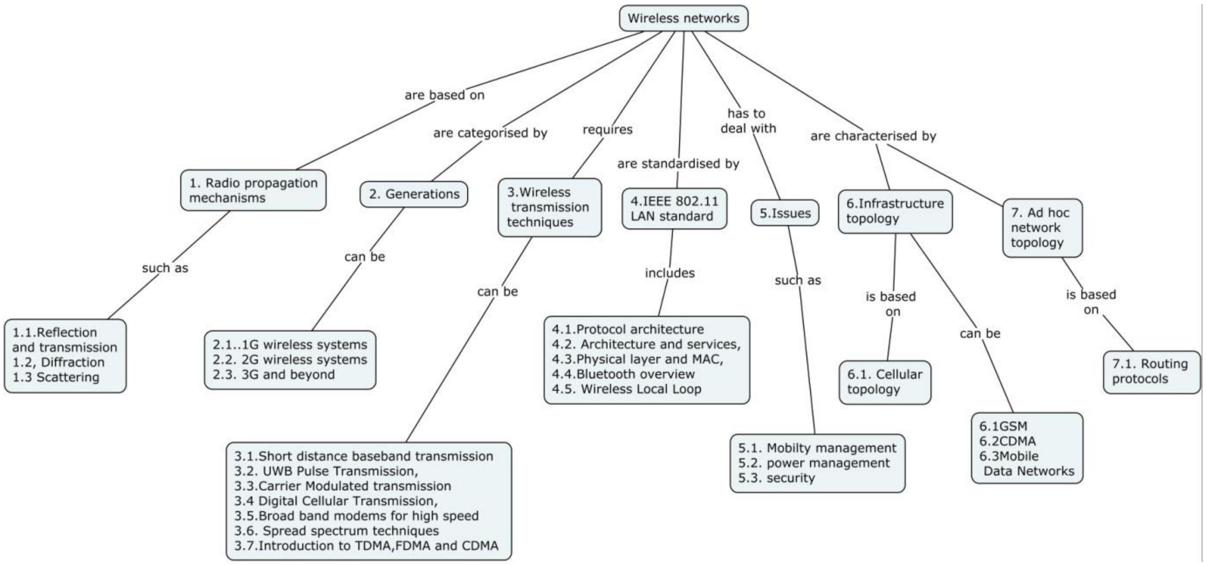
Course Outcome 5 (CO5)

1. State the need for Home agent and Foreign agent.
2. Define Anchor point**.**
3. A mobile has a home address of 136.142.117.21 and a care of address of 130.216.16.5. it listens to agent advertisement periodically . The agent advertisement indicates that the care of address is 130.216.45.3 what happens? Why?

Course Outcome 6 (CO6)

1. State the difference between secret key and public key algorithms.
2. Define Authentication**.**
3. A not so-rich hacker uses an old computer and brute force to break into some wireless systems. It takes him 1 ms on average to test a key to see if it is the right one for an encryption independent of the algorithm employed. How long it take him to break into IEEE 802.11 system in the worst case? How long will it take him to break into an IS-136 system on average ?

Concept Map



Syllabus

Radio propagation mechanisms-Reflection and transmission, Diffraction, Scattering **Generations of wireless networks-** 1G wireless systems, 2G wireless systems, 3G and beyond **Wireless transmission techniques** Short distance baseband transmission, UWB Pulse Transmission, Carrier Modulated transmission, Digital Cellular Transmission, Broad band modems for high speed, Spread spectrum techniques, Introduction to TDMA,FDMA and CDMA **IEEE 802.11 LAN standard-** Protocol architecture, Architecture and services, Physical layer and MAC, Bluetooth overview, Radio specification, Base band specification, Link manager specification ,logical link control, Wireless Local loop **Issues** mobility management ,power management ,security **Infrastructure Network Topology** Cellular topology ,Cell fundamentals, Signal to interference ratio calculation, Capacity expansion techniques, GSM, Mechanisms to support a mobile environment, CDMA - Comparison with GSM, Mobile data network, CDPD Network, GPRS, Short messaging services **Ad hoc network topology-**Introduction to Routing protocols

Text Books

1. Kaveh Pahlavan and Prashant Krishnamurthy, “Principles of Wireless Networks – A unified approach”, Pearson Education, Fourth Edition, 2003.
2. William Stallings, “Wireless Communications and Networks”, Pearson education, 2003

Reference Books

1. J. Schiller, “Mobile Communications”, Pearson education, 2003
2. C. Siva Ram Murthy and B.S.Manoj ,”Ad Hoc Wireless Networks Architectures and Protocols”, Pearson education, 2004.

3. Vijay K. Garg, “Wireless Communications and Networking”, Elsevier, 2008

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Radio propagation mechanisms(1)**	
1.1	Reflection and transmission	1
1.2	Diffraction	
1.3	Scattering	
2	**Generations of wireless networks(1)**	
2.1	1G wireless systems	1
2.2	2G wireless systems	
2.3	3G and beyond	
3	**Wireless transmission techniques(4)**	
3.1	Short distance baseband transmission	1
3.2	UWB Pulse Transmission	
3.3	Carrier Modulated transmission	1
3.4	Digital Cellular Transmission	
3.5	Broad band modems for high speed	1
3.6	Spread spectrum techniques	
3.7	Introduction to TDMA,FDMA and CDMA	1
4	**IEEE 802.11 LAN standard(6)**	
4.1	Protocol architecture	1
4.2	Architecture and services	1
4.3	Physical layer and MAC	1
4.4	Bluetooth overview	1
4.4.1	Radio specification	
4.4.2	Base band specification	

4.4.3	Link manager specification	1
4.4.4	Logical link control	
4.5.	Wireless Local Loop	1
5	**Issues(6)**	
5.1	Mobility management	2
5.2	Power management	2
5.3	Security	2
6	**Infrastructure Network Topology(16)**	
6.1	Cellular topology	1
6.1.1	Cell fundamentals	1
6.1.2	Signal to interference ratio calculation	
6.1.3	Capacity expansion techniques	2
6.1.4	Network planning for CDMA systems	1
6.2	GSM	1
6.2.1	Mechanisms to support a mobile environment	2
6.3	CDMA – Comparison with GSM	2
6.4	Mobile data network	1
6.4.1	CDPD Network	2
6.4.2	GPRS	2
6.4.3	Short messaging services	1
7	**Ad hoc network topology(2)**	
7.1	Introduction to Routing protocols	2
	Total	**36**

Course Designer

1. Mr. C.Senthilkumar cskcse@tce.edu

		Category	L	T	P	Credit
14CSPD0	**ENTERPRISE PROJECT DEVELOPMENT USING FOSS**	PE	3	0	0	3

Preamble

"Free/Open Source Software" is an excellent server platform, a good desktop, and the center of much innovation in the current world. It is discussed from the viewpoint of the application developer and from a practical perspective of using the operating system. In view of the growing base and popularity of the Open Source Linux, it is proposed to use it heavily.

Prerequisite

14CS420 – System Software and Operating Systems

Course Outcomes

On the successful completion of the course, students will be able to

Explain the internal components of GNU/Linux kernel architecture and their functions. (CO1)	Understand
Apply the standard tools like GNU Make, Autoconf, Automake to build packages in Free / Open Source Environment (CO2)	Apply
Identify the steps for setting up the web server for an enterprise using Linux deployment tools. (CO3)	Apply
Prepare posters and structured diagrams for the given requirement using Linux graphics tools.(CO4)	Apply
Apply utilities like sed and awk to find a pattern in a text. (CO5)	Apply
Describe the various version controlling systems and the licensing schemes. (CO6)	Understand

Mapping with Programme Outcomes

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	M											
CO2	S	M	L		S				M	M		L
CO3	S	M	L		S				M	M		L
CO4	S		L		S				M	M		L
CO5	S	M	L		M				L	L		L
CO6	M											

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	30	40	30	30
Apply	40	40	40	40
Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	10	0	10	10

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Review the GNU/Linux Operating system Architecture
2. Recall the timeline in history of Linux Operating System
3. Summarize the various optimization settings and descriptions

Course Outcome 2 (CO2):

1. Construct the following for generating all initial makefiles
    a) Autogen script file
    b) Makefile.am file
    c) Configure.ac file
2. Create a C Program using fread and fseek/rewind to read structured data?
3. Apply Pattern Matching rules to makefiles?
4. Illustrate Static, shared and dynamic Loaded Library with examples

Course Outcome 3 (CO3):

1. Recall the SSL Encryption
2. Discuss the Server Side Includes?
3. Define the CGI and how Apache specify what programs can be run as CGI Programs?
4. Create your own Apache Configuration file to administer multiple hosts?

Course Outcome 4 (CO4)

1. Using GIMP illustrate
    a) How to use GIMP Toolbox
    b) Change Image Size
    c) Remove elements from an image
    d) Apply layers to GIMP
2. Summarize the way to scan documents in Linux
3. Explain the means to move pictures from your camera to your computer

Course Outcome 5 (CO5)

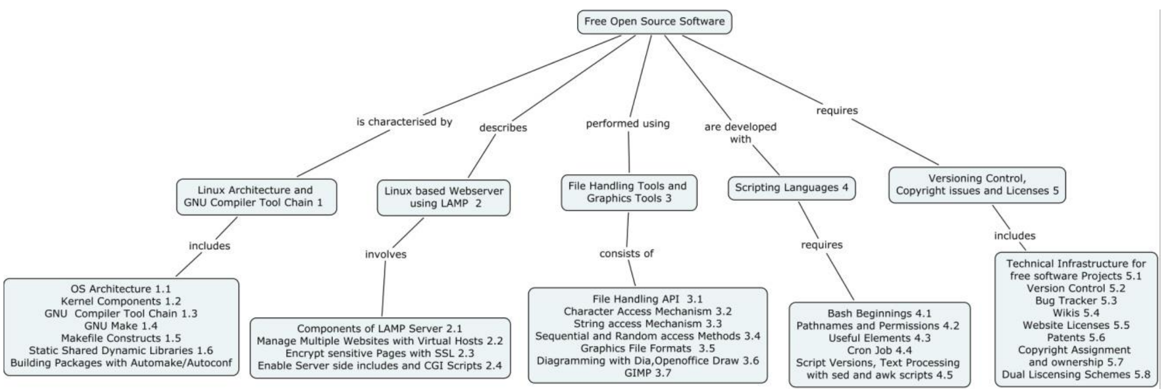
1. Explain the awk built-in-variables
2. Apply awk to find and store the extremes in a missile dataset

3. Define the Regular Expressions in sed?
4. Apply sed utility to
   a)Emit the first 10 lines of a file
   b)Emit all lines greater than 30 characters in length
   c)Emit all non-blank lines in a file
   d) Hold the pattern space

Course Outcome 6 (CO6)

1. Explain Version Control and its vocabulary with suitable examples
2. Discuss the Bug Tracker?
3. Explain the Copyright Issues and Licenses ?
4. Create a bash script to determine file attributes using file test operators ?

Concept Map



Syllabus

GNU/Linux Architecture and Development Tools: GNU/Linux Architecture, Architectural Breakdown of Major Kernel Components, Linux distributions, GNU Compiler Tool Chain, Building Software with GNU Make, Makefile Constructs. Static-Shared-Dynamic Libraries, Building packages with Automake/Autoconf. **Deployment Tools**: Components of a LAMP Server, Manage Multiple Websites with Virtual Hosts, Encrypt Sensitive Pages with SSL, Enable Server-side Includes and CGI Scripts; **File Handling Tools and Graphics Tools:** File Handling-API-Character access mechanisms, String access mechanisms, Sequential and Random access methods, Graphics File Formats, Diagramming with Dia, Open Office Draw, GIMP; **Text Processing Tools:** Bash beginnings, Pathnames and Permissions, Useful elements, cron Job, Script Versions Text Processing with awk and sed scripts; **Versioning Control, Copyright issues and licenses:** Standards for free software projects, Version Control, Bug Tracker, Wikis, Website Licenses, Patents, Copyright assignment and Ownership, Dual Licensing Schemes.

Text Books

1. M.Tim Jones, "GNU/Linux Application Programming", Dream Tech Press, 2005
2. Karl Fogel, "Producing Open Source Software", O'Reilly Media Inc, 2005
3. Janet Valade, "Spring into Linux", Pearson Education, 2006
4. Tom Adelstein and Bill Lubanovic, "Linux System Administration", O'Reilly, 2007.

Reference Books

1. I.Christopher Negus, " Linux Bible", Wiley, 2006.
2. Ellie Quigley, "PERL by Example", Pearson Education, 2009.

Course Contents and Lecture Schedule

S.No.	Topic	No. of Lectures
1	**GNU/Linux Architecture and Development Tools**	
1.1	GNU/Linux Architecture	1
1.2	Architectural Breakdown of Major Kernel Components, Linux distributions	1
1.3	GNU Compiler Tool Chain	1
1.4	Building Software with GNU Make	1
1.5	Makefile Constructs.	1
1.6	Static-Shared-Dynamic Libraries	1
	Building packages with Automake/Autoconf	1
2	**Deployment Tools**	
2.1	Components of a LAMP Server,	1
2.2	Manage Multiple Websites with Virtual Hosts,	1
2.3	Encrypt Sensitive Pages with SSL,	1
2.4	Enable Server-side Includes and CGI Scripts.	1
3	**File Handling Tools and Graphics Tools**	
3.1	File Handling-API-Character access mechanisms	1
3.2	String access mechanisms	2
3.3	Sequential and Random access methods	1
3.4	Graphics File Formats,	1
3.5	Diagramming with Dia, Open Office Draw	1
3.6	GIMP	1
4	**Text Processing Tools**	

S.No.	Topic	No. of Lectures
4.1	Bash beginnings	2
4.2	Pathnames and Permissions	1
4.3	Useful elements	1
4.4	Cron Job	2
4.5	Script Versions, Text Processing with awk and sed scripts	2
5	**Versioning Control, Copyright issues and licenses**	
5.1	Standards for free software projects	3
5.2	Version Control	1
5.3	Bug Tracker	1
5.4	Wikis	1
5.5	Website Licenses	1
5.6	Patents	1
5.7	Copyright assignment and Ownership	1
5.8	Dual Licensing Schemes	1
	Total	**36**

Course Designers

1. Dr. S. Mercy Shalinie smscse@tce.edu
2. Dr. K. Sundarakantham kskcse@tce.edu

14CSPE0	INFORMATION RETRIEVAL	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

Information retrieval is concerned with techniques, strategies, representations, and search basics of fetching information. It is expected to obtain specific facts, answer questions, or compose reports that enable students to understand and apply the principles for today's demanding information needs.

Prerequisite

14CS350 : Data Structures and Algorithms

14CS430 :Design and Analysis of Algorithms

Course Outcomes

On the successful completion of the course, students will be able to

Explain the principles of Information Retrieval (CO1)	Understand
Describe the pre-processing methods for Information Retrieval (CO2)	Understand
Apply appropriate compression techniques for dictionary files, posting files, and text data (CO3)	Apply
Examine the performance of IR system with various metrics like precision, recall and F-Measure.(CO4)	Analyze
Construct and parse XML documents for a given real time scenario (CO5)	Apply
Identify near duplicate documents by generating finger prints using Shingling approach (CO6)	Analyze
Compute the Hub scores and Authority scores for a given web graph (CO7)	Apply

Mapping with Programme Outcomes

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	M											
CO3.	S	M	L									
CO4.	S	S	M	L								
CO5.	S	M	L		M							L
CO6	S	S	M	L							L	L
CO7	S	M	L	L							L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	10	15
Understand	30	20	10	15
Apply	40	30	50	50
Analyze	-	30	30	20
Evaluate	-	--	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Write BSBI algorithm
2. Define Heaps Law
3. What is the need for cosine similarity vector?

Course Outcome 2 (CO2):

1. Draw the inverted index that would be built for the following document collection.
   i. Doc 1 : new scientific treatment
   ii. Doc 2: new cough drug
   iii. Doc 3: new approach for treatment of cough
2. Compare BSBI and SPIMI index construction methods
3. Compare dictionary as a string storage and blocked storage method for compression Techniques.

Course Outcome 3 (CO3):

1. Apply the Leveinstein edit distance between SONY and PONY
2. Compute variable byte code for Docids 210447 and 222335 also apply decoding algorithms.
3. Compute Huffman code for the following: “A canner can can anything that he can”

Course Outcome 4 (CO4)

1. a. Consider the table of term frequencies for 3 documents denoted OS, DB, and PP in Fig.1. Compute the Cosine similarity between them

Fig.1

Term	OS	DB	PP
code	110	50	20
execute	10	9	11
run	2	0	8

b. Find the score of the given values;

Term	tf	dft
Car Auto Insurance	25 30 40	18,111 6523 18,241

Total document collection: 22,222

2. Calculate the precision ,recall and MAP for the given table for total collection of 10 Documents. [N- no relevance, Y- relevance]

System1	N	Y	Y	N	Y	Y	Y	N	Y	N
System2	Y	N	N	Y	N	N	Y	N	Y	N

Course Outcome 5 (CO5):

1. Write XML for Student-course-subject--roll number- marks-result and Construct DOM tree for the above

2. a. Write XML document for the following scenario “Purchasing any 2 Books from book seller”

   b. Construct DOM tree for the above.

   c. Drive an Extended query from the above scenario

3. a. Write XML for –BANK-DEPOSIT-CREDIT –DEBIT-Transaction details.
   b. Construct DOM tree for the above
   c. Give X-Path example for the above

Course Outcome 6 (CO6):

1. Find the near duplicate for the table using Hamming distance problem .(F=Finger print certificate) . F=0100 1110 k= 4 bit

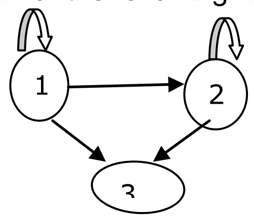
Finger Print
1100 1101
1111 1111
1100 0101
1110 0111
1011 1001
0100 1111
1011 0100
0101 1110

2. Find the certificate F for the following" rose is a rose is a rose" (apply ASCII for characters)

Course Outcome 7 (CO7):

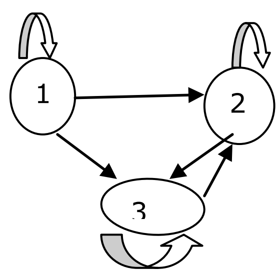
1. Consider a web graph with three nodes 1,2 and 3.The links are as follows : 1 to 2, 3 to 2, 2 to 1, 2 to 3. Write down the transition probability matrices for the surfers walk with teleporting value =0.4 and find the page rank vector.

2. Find Page rank for the following web graph: α= 0.4

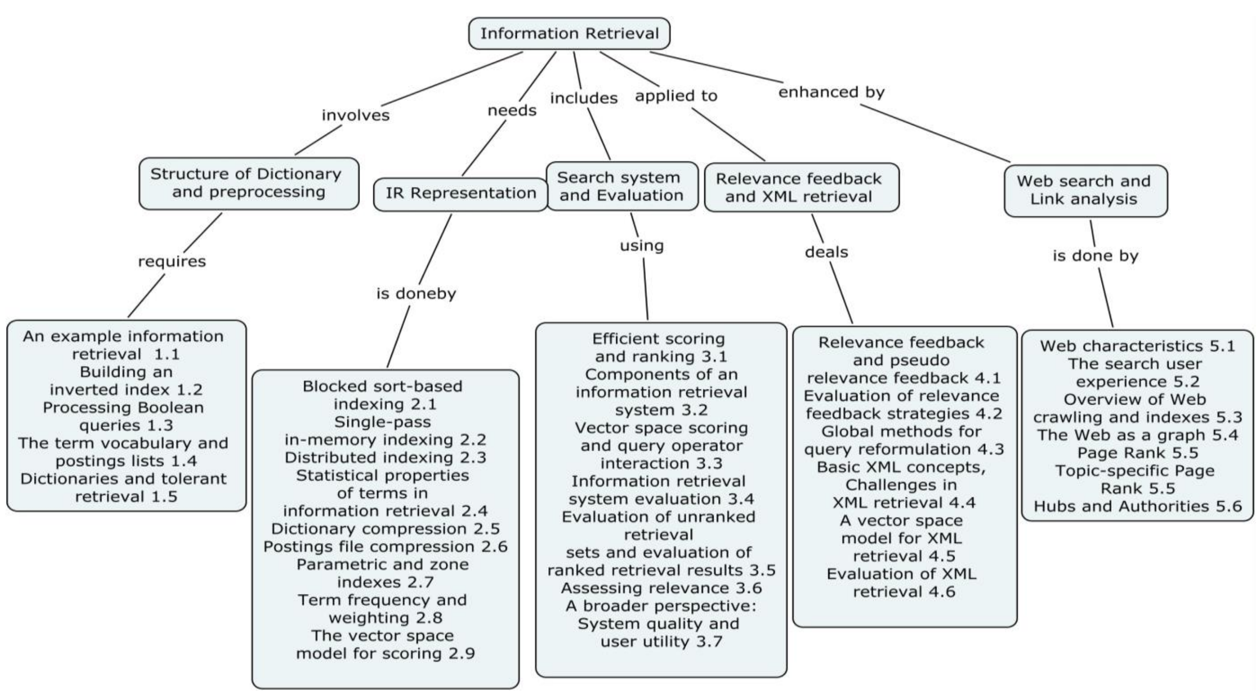


3 . Find hub score and authority score for the following graph

.



Concept Map



Syllabus

Structure of Dictionary and preprocessing: An example information retrieval problem , Processing Boolean queries, the extended Boolean model versus ranked retrieval, document delineation and character sequence decoding, obtaining the character sequence in a document determining the vocabulary of terms, tokenization, normalization, Stemming and lemmatization , search structures for dictionaries, general wildcard queries-gram indexes for wildcard queries, k-gram indexes for spelling correction, faster postings list intersection via skip pointers ,positional postings and phrase queries. **IR Representation** : hardware basics, blocked sort-based indexing ,single-pass in-memory indexing, distributed indexing, dynamic indexing, statistical properties of terms in information retrieval,heaps' law,Zipf's law, dictionary compression, Text data compression-Huffman tree method parametric and zone indexes, weighted zone scoring, learning weights, term frequency and weighting, inverse document frequency,Tf-idf weighting, the vector space model for scoring. Variant tf-idf functions. **Search system and Evaluation:** efficient scoring and ranking ,inexact top K document retrieval, index elimination ,champion lists ,static quality scores and ordering, components of an information retrieval system ,tiered indexes, query-term proximity. Information retrieval system evaluation, standard test collections, evaluation of unranked retrieval sets, evaluation of ranked retrieval results, assessing relevance, a broader perspective: System quality and user utility, system issues. **Relevance feedback and XML retrieval:** Relevance feedback and pseudo relevance feedback ,the Rocchio algorithm for relevance feedback ,probabilistic relevance feedback ,relevance feedback on the web, evaluation of relevance feedback strategies, basic XML concepts, challenges in XML retrieval, a vector space model for XML retrieval, evaluation of XML retrieval, text-centric vs. data-centric XML retrieval **Web search and Link analysis:** Web search basics, web characteristics, the web graph, the search user experience, web crawling and indexes, crawler architecture. Finding Near duplicates ,,PageRank,the Page Rank computation, topic-specific PageRank,hubs and authorities

Text Book

1. Christopher D.Manning,Prabhakar Raghavan and Hinrich Schütze, "An Introduction to Information Retrieval", Cambridge University Press,Cambridge, England,2009

Reference Books

1. David A. Grossman, Ophir Frieder," Information Retrieval: Algorithms and Heuristics ,Springer (2nd Edition),2004

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
	Structure of Dictionary and pre processing (6)	
1.1	An example information retrieval problem	1
1.2	Building an inverted index	1
1.3	Processing Boolean queries	1
1.4	The term vocabulary and postings lists	1
1.5	Dictionaries and tolerant retrieval	2
	IR Representation (8)	
2.1	Blocked sort-based indexing	1
2.2	Single-pass in-memory indexing	1
2.3	Distributed indexing	1
2.4	Statistical properties of terms in information retrieval	1
2.5	Dictionary compression	1
2.6	Postings file compression	1
2.7	Text data Compression-Huffman tee algorithm	1
2.8	Parametric and zone indexes	1
	Search system and Evaluation (7)	
3.1	Efficient scoring and ranking	1
3.2	Components of an information retrieval system	1
3.3	Vector space scoring and query operator interaction	1
3.4	Information retrieval system evaluation	1
3.5	Evaluation of unranked retrieval sets and evaluation of ranked retrieval results	1
3.6	Assessing relevance	1
3.7	A broader perspective: System quality and user utility	1
	Relevance feedback and XML retrieval (6)	

4.1	Relevance feedback and pseudo relevance feedback	1
4.2	Evaluation of relevance feedback strategies	1
4.3	Global methods for query reformulation	1
4.4	Basic XML concepts, Challenges in XML retrieval	1
4.5	A vector space model for XML retrieval	1
4.6	Evaluation of XML retrieval	1
Web search and Link analysis (9)		
5.1	Web characteristics	1
5.2	Overview of Web crawling and indexes	1
5.3	The Web as a graph	1
5.4	Finding Near Duplicates-Shingling method	2
5.5	Page Rank	1
5.6	Topic-specific Page Rank	1
5.7	Hubs and Authorities	2
	Total	**36**

Course Designer

1. Dr. C.Deisy cdcse@tce.edu

14CSPF0	PARALLEL COMPUTING	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course will facilitate the Students to analyze and identify the hot spots in a serial application program. Students will learn the principles and techniques for programming the wide variety of parallel platforms currently available; able to design, develop and build a parallel processing architecture machine capable of executing logic programs. They will also learn how to solve the computational intensive applications in the cluster as well as in the cloud environment.

Prerequisite

- 14CS430 :Design and Analysis of Algorithms
- 14CS540 :Computer Architecture
- Basic understanding of analysis of algorithms and internal functionalities of a computer is needed.

Course Outcomes

On the successful completion of the course, students will be able to

Explain the fundamentals of parallel programming platforms, principles of design and parallel algorithm models (CO1)	Understand
Analyze the time complexity of PRAM computational approach for an application by varying the number of processors (CO2)	Analyze
Develop parallel algorithms for executing applications in a cluster computer with appropriate mapping and scheduling techniques (CO3)	Apply
Develop parallel algorithms for various applications by using domain decomposition techniques and parallel models (CO4)	Apply
Analyze the complexities involved in searching for an optimal solution for intelligent games (CO5)	Analyze

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	S	M	L									
CO3.	S	M	M	L								L
CO4.	S	M	M	L					L	L		L
CO5.	S	S	M	L					L	L		L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	10	10
Understand	20	20	10	10
Apply	30	40	40	40
Analyze	30	20	40	40
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Classify computers based on Flynn's Taxonomy.
2. Explain the parameters that characterize the processor organization.
3. Distinguish Data Decomposition from Exploratory decomposition.
4. Explain the characteristics of Inter task interaction.

Course Outcome 2 (CO2):

1. Analyze the time complexity of PRAM Algorithm for merging two sorted list of n/2 elements using appropriate number of processors.
2. Devise a PRAM algorithm to find prefix sum of 'n' elements using appropriate processors and estimate its time complexity.
3. Write Brent's theorem and prove it with suitable PRAM algorithm.
4. List out the phases of a PRAM algorithm.
5. Analyze the time complexity of EREW PRAM Algorithm for finding the sum of 'n' elements using n/2 processors.

Course Outcome 3 (CO3)

1. How to map a 8-node ring into a Cube?
2. Illustrate the mapping of data to processors on processor arrays with neat schematic diagram.
3. Devise static scheduling algorithms for UMA Multiprocessors with a neat Gantt chart.
4. Identify the time complexity of parallel summation algorithm consists of 16 elements On 8-Processors shuffle-exchange network with suitable schematic representation.
5. Illustrate the static mapping techniques based on data partitioning with suitable Examples.
6. Devise a Parallel algorithm for finding the sum of 16 values on a processor array organized as a 2-D Mesh
7. Analyze the time complexity of matrix multiplication on 2-D Mesh SIMD model with 4 x 4 matrix size. Assume enough number of processors for implementation

Course Outcome 4 (CO4)

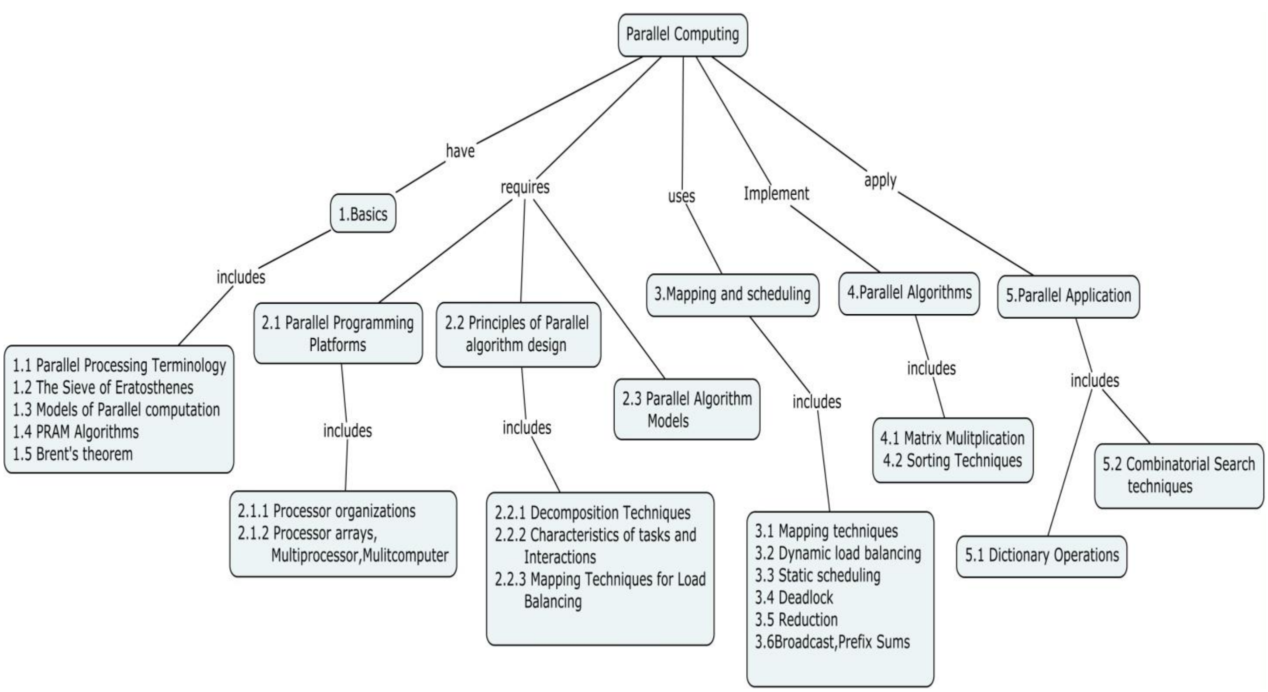
1. Examine various parallel algorithm models and figure out their significance and Applicability
2. Construct the task dependency graph for Quicksort algorithm to sort a sequence of 14 numbers using suitable decomposition technique.
3. Identify the time complexity of Bitonic Mergesort on the shuffle-exchange network using Stone's theorem.
4. Apply data parallel approach to the Sieve of Eratosthenes and Estimate the total communication time to sieve all 'k' primes.
5. Develop a Matrix multiplication algorithm for Shuffle Exchange network.
6. Implement the Hyperquick sort on the following sequence : No.of.Processors = 4

n = { 75,91,15,64,21,8,88,54,50,12,47,72,65,54,66,22,83,66,67,0,
70,98,99,82,20,40,89,47,19,61,86,85}

Course Outcome 5 (CO5)

1. Analyze the alpha beta search algorithm and evaluate the complexities in subtree generation for chess game tree.
2. Find the optimal solution for 8 Puzzle problems using branch and bound search technique and generate its state space tree corresponding to the search technique.
3. Illustrate Alpha Beta Pruning technique with suitable Game Tree.
4. Write about the importance of the combinatorial search.
5. Apply Best First Branch and Bound search for Traveling salesperson problem for find an optimal tour and represent it in state space tree.

Concept Map



Syllabus

Basics: Introduction, Parallel processing terminology, The Sieve of Eratosthenes, Model for parallel computation, PRAM Algorithms, Brent's Theorem. **Parallel Programming Platforms, Design Principles and Models:** Processor organization, Processor array, Multiprocessor and Multicomputer, Flynn's Taxonomy, Principles of Parallel Algorithm Design, Decomposition Techniques, Characteristics of tasks and Interactions, Mapping Techniques for Load Balancing, Parallel Algorithm Models. **Mapping and Scheduling:** Mapping data to processors on processor arrays and Multicomputers, Static Scheduling on UMA Multiprocessors, Deadlock, Elementary Parallel Algorithms, Classifying MIMD Algorithms, Reduction, Broadcast, Prefix Sums. **Parallel Algorithms:** Matrix Multiplication, Sequential Algorithm, Algorithm for Processor Array, Algorithms for Multicomputer and for Multiprocessors, Sorting, Enumeration Sort, Odd even transposition sort, Bitonic Merge sort, Quick Sort based Algorithms. **Parallel Applications:** Dictionary operations, Complexity of Parallel Search, Searching on Multiprocessors, Combinatorial Search, Divide and Conquer, Branch and Bound, Parallel Branch and Bound Algorithms, Alpha Beta Search, Parallel Aplha Beta Search.

Text Book

1.Michael J. Quinn, "Parallel Computing –Theory and Practice", Tata McGraw Hill Publishing Company Limited, Second edition, 2006.

Reference Books

1.Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, " Introduction to Parallel Computing", Pearson Education, Second Edition, 2009.

2.Calvin Lin, Lawrence Snyder, "Principles of Parallel Programming", Pearson Education, First Edition, 2010.

3.V.Rajaraman, C.Sivaraman, "Parallel Computers – Architecture and Programming", Prentice Hall of India, 2009.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Basics (5)**	
1.1	Parallel Processing terminology	1
1.2	The sieve of Eratosthenes	1
1.3	Models for parallel computation	1
1.4	PRAM Algorithms	1
1.5	Brent's Theorem	1
2.	**Parallel Programming – Platforms, Principles and Models (6)**	
2.1.1	Processor Organizations	1

Module No.	Topic	No. of Lectures
2.1.2	Processor arrays, Multiprocessor, Multicomputers	1
2.2	Principles of Parallel Algorithm Design	1
2.2.1	Decomposition Techniques	1
2.2.2	Characteristics of tasks and Interactions	1
2.2.3	Mapping Techniques for Load Balancing	1
2.3	Parallel Algorithm Models	1
3	**Mapping and scheduling (7)**	
3.1	Mapping data to processors on processor arrays and multicomputers	1
3.2	Dynamic load balancing on multicomputers	1
3.3	Static scheduling on UMA multiprocessor	1
3.4	Deadlock	1
3.5	Reduction	2
3.6	Broadcast,Prefix Sums	1
4	**Parallel Algorithms (9)**	
4.1	Matrix multiplication-Sequential algorithms	1
4.1.1	Algorithms for processor array	2
4.1.2	Algorithms for multicomputers & multiprocessors	1
4.2	Sorting	
4.2.1	Enumeration sort, Odd Even Transposition sort	1
4.2.2	Bitonic merge	2
4.2.3	Quick sort –based algorithms	2
5	**Parallel Applications (9)**	
5.1	Dictionary operations-Complexity of parallel search	2
5.1.1	Searching on multiprocessors	1
5.2	Combinatorial search-Divide and Conquer	1
5.2.1	Branch and Bound	1
5.2.2	Parallel Branch and Bound algorithms	1
5.2.3	Alpha beta search	2
5.2.4	Parallel Alpha Beta Search	1
	Total No.of Hours	36

Course Designers

1. Dr. S. Mercy Shalinie smscse@tce.edu
2. Dr. S. Padmavathi spmcse@tce.edu

14CSPG0	STORAGE INFRASTRUCTURE MANAGEMENT	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

The course on Storage Infrastructure Management aims to emphasize the need for information storage, provide an in depth coverage of technologies in the various phases of designing and building an Information Storage System and to provide an overview of various management techniques.

Prerequisite

- 14CS440 : Database Management Systems
- 14CS520 : Computer Networks

Course Outcomes

On the successful completion of the course, students will be able to

Explain the components and functions of information storage systems. (CO1)	Understand
Describe the functionalities of Storage Networking .(CO2)	Understand
Illustrate the process of Backup and Replication (CO3)	Understand
Develop the storage system for the given specification.(CO4)	Apply
Compute the performance metrics of storage components.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	M											
CO3	M											
CO4	S	M	M									
CO5	S	L										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	60	40	40	40
Apply	20	40	40	40
Analyze	0	0	0	0

Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Mention the categories of data.
2. What do you mean by downtime?
3. List the demerits of centralized data storage.
4. Define Platter.
5. Describe how you can control Application access, User access and Host access.

Course Outcome 2 (CO2):

1. List the challenges of NAS.
2. Define fixed content.
3. Explain how CAS stores and retrieves data objects.
4. Mention the benefits of CAS.
5. Describe the functionalities of FC Switch and Hub with necessary diagrams.

Course Outcome 3 (CO3):

1. Explain how remote replication technology can be helpful in disaster recovery.
2. Write the use of Backup?
3. Classify Replication.
4. Define RPO and RTO.
5. Differentiate Disaster Recovery and Disaster Restart.

Course Outcome 4 (CO4):

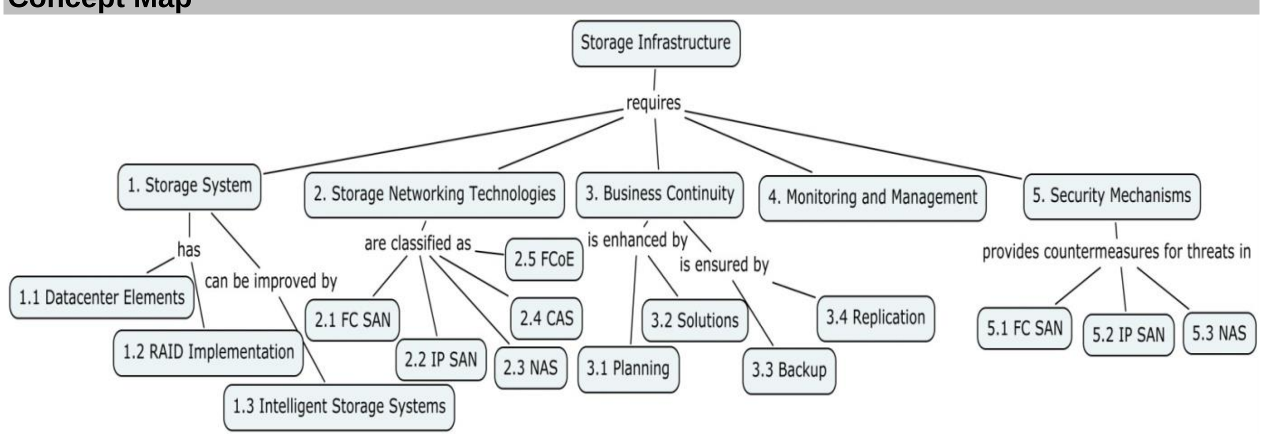
1. ABC Corporation is trying to decide between an integrated or a gateway NAS solution. The existing SAN at ABC will provide capacity and scalability. The IT department is considering a NAS solution for the training department at ABC for training videos. The videos will only be used by the training department for evaluation of instructors. Develop a NAS solution.
2. ABC Corporation is trying to decide between an integrated or a gateway NAS solution. The existing SAN at ABC will provide capacity and scalability. The IT department is considering a NAS solution for the training department at ABC for training videos. The videos will only be used by the training department for evaluation of instructors. Suggest a NAS solution.
3. A company is considering storage implementation. They do not have a current storage infrastructure to use, but they have a network that gives them good performance. Suggest whether native or bridged iSCSI should be used.

Course Outcome 5 (CO5) :

1. An application specifies a requirement of 200GB to host a database and other files. It also specifies that the storage environment should support 5000 IOPS during its peak processing cycle. The disks available for configuration provide 66GB of usable capacity and the manufacturer specifies that they can support a maximum of 140 IOPS. The application is response time sensitive and the disk utilization beyond 60% will not meet the response time requirements of the application. Compute the minimum number of disks that should be configured to meet the requirements of the application.
2. Consider a disk I/O system in which an I/O request arrives at the rate of 80 IOPS. The disk service time is 6ms.
   a. Compute the following
      i. Utilization of I/O controller
      ii. Total response time
      iii. Average queue size
      iv. Total time spent by a request in a queue

   Compute the preceding parameter if the service time is halved.
3. A 10k RPM drive is rated to perform 130 IOPS and a 15k RPM drive is rated to perform 180 IOPS for an application. The read/write ratio is 3:1. Compute the RAID-adjusted IOPS for the 10k and 15k drives for RAID 1, RAID 5 and RAID 6.
4. An application has 1000 heavy users at a peak of 2 IOPS each and 2000 typical users at a peak of 1 IOPS each, with a read/write ratio of 2:1. It is estimated that the application also experiences an overhead of 20% for other workloads. Calculate the IOPS requirement for RAID 1, RAID 3, RAID 5 and RAID 6.

Concept Map



Syllabus

Storage System: Introduction - Evolution of storage architecture - Key Datacenter elements – Host, connectivity, storage, and application in both classic and virtual environments - RAID implementations – techniques - RAID levels - impact of RAID on application performance - Components of Intelligent Storage Systems - Provisioning and Intelligent Storage System **Storage Networking Technologies:** Fibre Channel SAN - components - Connectivity options - topologies - Access protection mechanism – zoning - FC protocol stack – Addressing - SAN-based virtualization – VSAN - IP SAN - iSCSI and FCIP protocols for Storage access over IP network - FCoE and its components - Network Attached Storage (NAS) – NAS Hardware devices – NAS Software Components – NAS Connectivity options - NAS operations – Applying the NAS Solution – File level virtualization in NAS – Integration of NAS and SAN - CAS –Object based storage - Unified Storage platform. **Business Continuity:** Information availability and Business Continuity - Business Continuity terminologies - Business Continuity Planning – Solutions - Clustering and Multipathing architecture - Single Points of Failure - Backup and Recovery - Methods, targets and topologies - Data Deduplication and backup in virtualized environment - Fixed Content and Data Archive – Replication - Local Replication - Remote Replication - Three-Site Remote Replication - Continuous Data Protection **Monitoring and Management:** Monitoring and managing storage infrastructure components in classic and virtual environments - Information lifecycle management (ILM) and Storage Tiering - Cloud service management **Security Mechanisms:** Security threats and countermeasures in various domains – Security solutions for FC-SAN, IP-SAN and NAS environments - Security in virtualized and cloud environments

Text Book

1. Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments, EMC Education Services, John Wiley and Sons, 2012, ISBN: 9781118094839

Reference Books

1. “Storage Networks: The Complete Reference”, Robert Spalding, Tata McGraw Hill-Osborne, 2003.
2. “Building Storage Networks”, Marc Farley, Tata McGraw Hill-Osborne, 2001.
3. “Disaster Recovery and Business Continuity”, Thejendra BS, Shroff Publishers and Distributors, 2006.
4. Storage Virtualization, Clark Tom, Addison Wesley Publishing Company, 2005, ISBN : 9780321262516

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Storage System**	
1.1	Datacenter Elements - Evolution of storage architecture	2
1.2	RAID implementation - techniques - RAID levels - impact of RAID on application performance	2
1.3	Intelligent Storage Systems – Components - Provisioning and Intelligent Storage System	2
2	**Storage Networking Technologies**	
2.1	FC SAN - Components - Connectivity options - Topologies – Zoning - FC protocol stack – Addressing - SAN-based virtualization	2
2.2	IP SAN - iSCSI and FCIP protocols for Storage access over IP network	3
2.3	NAS - Components, Connectivity, Protocol and Operations - File level virtualization – Integration of NAS and SAN	3
2.4	CAS - Object based storage	2
2.5	FCoE - Components	2
3	**Business Continuity**	
3.1	Business Continuity Planning - Single Points of Failure - Information availability	2
3.2	Business Continuity Solutions - Clustering and Multipathing architecture	2
3.3	Backup and Recovery - Methods, targets and topologies - Data Deduplication and backup in virtualized environment	3
3.4	Replication – Local and Remote, Three-Site Remote Replication	2
4	**Monitoring and Management**	3
5	**Security Mechanisms**	
5.1	Security Mechanisms for FC SAN, VSAN	2
5.2	Security Mechanisms for IP SAN	2
5.3	Security Mechanisms for NAS	2
	Total No. of hours	36

Course Designers:

1. Dr.G.S.R. Emil Selvan emil@tce.edu
2. Mr.M.P.Ramkumar ramkumar@tce.edu

14CSPH0	SERVICE-ORIENTED ARCHITECTURE	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

Service-Oriented Architecture (SOA) is a revolutionary computing platform that is being adopted world-wide and has earned the support of every major software provider. It is the main approach for dealing with the interoperability of systems in heterogeneous environments. The student obtains clear understanding of what constitutes SOA along with step-by-step guidance for realizing its successful implementation. The Student will be able to apply SOA principles to real time needs and develop enterprise applications using those principles.

Prerequisite

- 14CS520: Computer Networks
- 14CS620: Internet Programming

Course Outcomes

On the successful completion of the course, students will be able to

Describe the characteristics, benefits and evolution of Service Oriented Architecture (CO1)	Understand
Outline the functions and design principles of the different layers in a Service Oriented Architecture and their inter relationships(CO2)	Understand
Develop web services with an understanding of the underlying SOA principles like service descriptions, messaging, activity management, composition, interoperability and security. (CO3)	Apply
Construct service oriented business process design for an enterprise by using Web Service-Business Process Execution Language.(CO4)	Apply
Implement SOA designs for an enterprise on top of platform technologies like .NET and J2EE. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	M											
CO3.	S	M	M									
CO4.	S	M	M	L				L	L	L		L
CO5.	S	M	M	L	S			L	L	L		L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	40	20	20	20
Understand	40	40	20	20
Apply	20	40	60	60
Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List down the benefits of SOA.
2. Discuss about Application architecture, Enterprise architecture and Service Oriented Architecture.
3. Explain Service-Oriented Architecture and Client-Server architecture.
4. Write about the basic components that defines SOA as an architecture model?
5. What is contemporary SOA?

Course Outcome 2 (CO2):

1. How the services are reusable?
2. Distinguish autonomy and statelessness.
3. List out the components of SOA and their relationship.
4. Identify the common principles of service orientation.
5. Compare SOA Principles with Object orientation principles in terms of service contract, reusability, abstraction and discoverability.

Course Outcome 3 (CO3)

1. Develop web services for a car washing company handling administrative tasks deploying SOA principles like endpoint references and message information headers.
2. How can we increase SOA's quality of service level?
3. Write a WS-Policy for an enterprise maintaining Vendor Profile Service which will accept invoice submission sequence headers that conforms to both versions of WS-Reliable Messaging.
4. Develop web services with an understanding of the underlying SOA principles like service descriptions, messaging, activity management, composition, interoperability and security for a online ticket reservation system.
5. Develop a web service for an enterprise handling online shopping to receive notifications from the Logistics Company notification service.

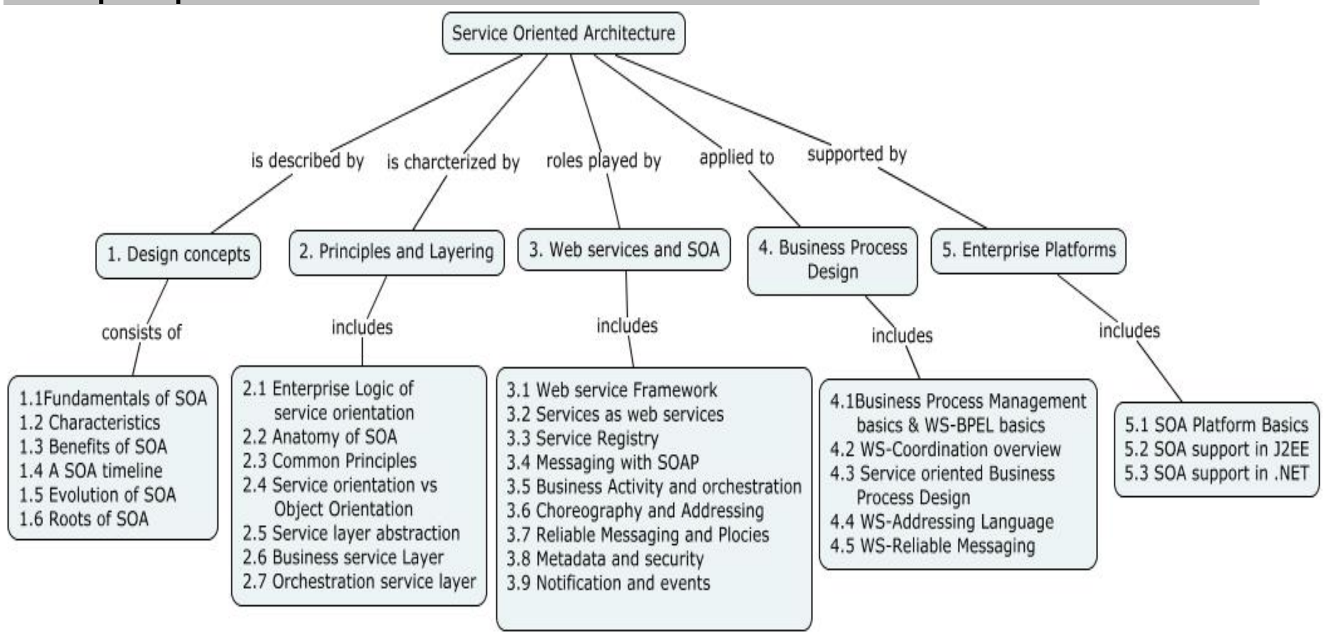
Course Outcome 4 (CO4)

1. Illustrate the use of WS-Addressing Specification in SOA.
2. Illustrate the step by step design of WS-BPEL Process Definition.
3. Construct service oriented business process design for an enterprise by using WS-BPEL.
4. Construct a high level service oriented business process design for an enterprise which has Timesheet submission Process as a service by using WS- BPEL

Course Outcome 5 (CO5)

1. Make use of J2EE technologies work together within the service requestor model for performing variety of runtime filtering, processing and routing tasks for an enterprise.
2. Explain how ASP.NET Web Forms can be used to build the presentation layer of a service oriented solution.
3. How parts of the .NET framework inter-relate?
4. Illustrate how .NET proxies behave within the standard service requestor model.
5. Illustrate how J2EE support contemporary SOA principles.

Concept Map



Syllabus

Introduction to SOA and Evolution of SOA: Fundamental SOA, Common Characteristics of contemporary SOA, Benefits of SOA, A SOA timeline, The continuing evolution of SOA, The roots of SOA. **Principles of Service–Orientation and Service Layer**: Services-orientation and the enterprise, Anatomy of a service-oriented architecture, Common Principles of Service-orientation, Service orientation and Object-orientation, Service layer abstraction, Business service layer, Orchestration service layer. **Web Services and SOA:** The Web services framework, Services, Service Registry, Service descriptions, Messaging with Simple Object Access Protocl, Transactions, Coordination, Business Activity, Orchestration, Choreography, Addressing, Reliable Messaging, Policies, Metadata, Security, Notification and Events; **Business Process Design**: Business Process Management basics, WS-BPEL language basics, WS-Coordination overview, Service oriented business process design, WS-addressing language basics, WS-Reliable Messaging language basics. **Enterprise Platforms and SOA**: SOA platform basics, SOA support in J2EE, SOA support in .NET.

Text Book

1. Service-Oriented Architecture: Concepts and Technology and Design by Thomas Erl, Pearson Education, 2005.

Reference Books

1. Understanding SOA with Web Services – Eric Newcomer, Greg Lomow, Pearson Education, 2005.
2. Developing Enterprise Web Services – An Architect's Guide – Sandeep Chatterjee, James Webber Pearson Education.

Websites

- Patterns: Service Oriented Architecture and Web Services: http://www.redbooks.ibm.com/abstracts/sg246303.html?Open
- IBM developerWorks Web Services Zone http://www-128.ibm.com/developerworks/websphere/zones/webservices/
- SOA Reference Architecture: http://www.ibm.com/developerworks/library/ar-archtemp/ http://www.soablueprint.com/reference_architecture
- Service Component Architecture: http://www.osoa.org/display/Main/Service+Component+Architecture+Home
- Architectural Styles and the Design of Network-based Software Architectures: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Course Contents and Lecture Schedule

S.No.	Topic	No. of Lectures
1	**Introduction to SOA and Evolution of SOA (6 hours)**	
1.1	Fundamental SOA	1
1.2	Common Characteristics of contemporary SOA	1
1.3	Benefits of SOA	1
1.4	A SOA timeline	1
1.5	The continuing evolution of SOA	1
1.6	The roots of SOA	1
2	**Principles of Service-Orientation and Service Layer (7 hours)**	
2.1	Services-Orientation and The Enterprise	1
2.2	Anatomy of a Service-Oriented Architecture	1
2.3	Common Principles of Service-Orientation	1

S.No.	Topic	No. of Lectures
2.4	Service-Orientation and Object-Orientation	1
2.5	Service Layer Abstraction	1
2.6	Business Service Layer	1
2.7	Orchestration Service Layer	1
3	**Web Services and SOA (9 hours)**	
3.1	The Web Services Framework - Overview	1
3.2	Services as Web Services	1
3.3	Service Registry and Service Descriptions	1
3.4	Messaging with SOAP, Transactions and Coordination	1
3.5	Business Activity and Orchestration	1
3.6	Choreography and Addressing	1
3.7	Reliable Messaging and Policies	1
3.8	Metadata and Security	1
3.9	Notification and Events, RESTful Web services: The basics	1
4	**Business Process Design (8 hours)**	
4.1	Business Process Management Basics, WS-BPEL language basics	2
4.2	WS-Coordination overview	1
4.3	Service Oriented Business Process Design	2
4.4	WS-Addressing Language basics	1
4.5	WS-Reliable Messaging Language Basics	2
5	**Enterprise Platforms and SOA (6 Hours)**	
5.1	SOA Platform Basics	2
5.2	SOA support in J2EE	2
5.3	SOA support in .NET	2
	Total	**36**

Course Designers:

1. Dr. S. Padmavathi spmcse@tce.edu

14CSPJ0	**DESIGN AND ANALYSIS OF ALGORITHMS - II**	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

Algorithm design and analysis is a fundamental and important part of computer science. This course introduces students to advanced techniques for the design and analysis of algorithms, and explores a variety of applications. In this paper, survey of many techniques that can be apply broadly in the design of efficient algorithms, and study their application in a wide range of application domains and computational models.

Prerequisite

- Data Structures and Algorithms
- Design and Analysis of Algorithms

Course Outcomes

On the successful completion of the course, students will be able to

Solve string matching problems using string algorithms including Rabin-Karp Fingerprinting Algorithm and Suffix Trees (CO1)	Apply
Solve maximum flow problems by finding a feasible flow through a single-source, single-sink flow network that is maximum (CO2)	Apply
Formulate the real world problems as linear programming problems and solve linear programming using polynomial time algorithms (CO3)	Apply
Solve real world decision making problems using online algorithms including Ski Rental, River Search Problem, Paging, The k-Server Problem, List Ordering and Move-to-Front (CO4)	Analyze
Demonstrate approximation algorithms for NP-hard problems such as set cover and vertex cover problems (CO5)	Analyze
Solve Computational Geometry problems Convex Hull, Line-segment Intersection, Sweep Lines, Voronoi Diagrams, Range Trees, Seidel's Low-dimensional LP Algorithm (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	S	L	L	M							L
CO2.	S	S	L	L	M							L
CO3	S	S	L	L	M							L
CO4	S	S	L	L	M							L
CO5	S	S	L	L	M							L
CO6	S	S	L	L	M							L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	10	10	10	10
Understand	10	10	10	10
Apply	80	60	60	60
Analyse	0	20	20	20
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Given two strings s and t, compute is the longest substring that occurs in both of them. For example if a="boogie" and b="ogre" then the answer is "og".
2. Given a string s, find the longest substring that is a palindrome (or a Watson-crick palindrome).
3. How would you modify Rabin-Karp to search for a given pattern with the additional proviso that the middle character is a "wildcard" (any text character at all can match it).
4. How would you modify Rabin-Karp to determine whether any of a subset of k patterns (say, all of the same length) is in the text?
5. Online palindrome detection. Read in characters one at a time. Report at each instant if the current string is a palindrome.

Course Outcome 2 (CO2):

1. The transportation problem is a generalization of the minimum-cost bipartite matching problem. You are given a bipartite graph G = (A ∪ B, E), edge costs c : E → N, supplies s : A → N≥0, and demands d : B → N≥0, such that the total supply equals the total demand.

$$\sum_{a \in A} s(a) = \sum_{b \in B} d(b) = U$$

   A feasible solution is a transportation f : E → $N_{\geq 0}$ of supplies to demands, i.e.:

$$\begin{aligned} \sum_{b \in B} f(a,b) &= s(a) \qquad \forall a \in A \\ \sum_{a \in A} f(a,b) &= d(b) \qquad \forall b \in B \end{aligned}$$

   The cost of a transportation f is:

$$c(f) = \sum_{e \in E} c(e) f(e)$$

   The transportation problem asks to find a transportation of minimum cost.
   (a) Show that transportation is a special case of minimum-cost flow.
   (b) Give an O(|V | + |E|) time reduction from minimum-cost flow to the transportation problem. That is, describe the transformation from a min-cost flow instance to a transportation instance, and show how a transportation solution can be bijectively mapped to a min-cost flow solution of the same value.

(Hint: The transportation problem has no edge capacities. How can the MCF capacity constraints be expressed as transportation supply/demand constraints?)

2. An s-t vertex cut is defined as a subset of vertices whose removal disconnects vertices s and t. Given a unit-capacity, undirected graph G and two vertices s and t, we want to show that the maximum number of vertex-disjoint s-t paths is equal to the minimum size of an s-t vertex cut. To show this, formulate the problems as a primal-dual LP pair and show that both LPs have integral optimal solutions.
3. Let G be a directed, unit-capacity graph with n vertices and m edges, where every vertex (beside s, t) has in-degree or out-degree at most 1. Show that $O(n^{1/2})$ blocking flows suffice in finding a maximum flow.
4. A matching in an undirected graph is a subset of edges no two of which share an endpoint. Use the property you derived in previous question to obtain an $O(mn^{1/2})$-time algorithm for finding a matching of maximum cardinality in an undirected bipartite graph.
5. Given a unit-capacity directed graph with n vertices and m edges, show that $O(n^{2/3})$ blocking flows suffice in finding a maximum flow. Now, use this property to obtain an $O(mn^{2/3})$-time maximum flow algorithm for unit-capacity directed graphs.

Course Outcome 3 (CO3):

1. Suppose that a farmer has a piece of farm land, say L km2, to be planted with either wheat or barley or some combination of the two. The farmer has a limited amount of fertilizer, F kilograms, and insecticide, P kilograms. Every square kilometer of wheat requires F1 kilograms of fertilizer and P1 kilograms of insecticide, while every square kilometer of barley requires F2kilograms of fertilizer and P2 kilograms of insecticide. Let S1 be the selling price of wheat per square kilometer, and S2 be the selling price of barley. If we denote the area of land planted with wheat and barley by x1 and x2 respectively, then profit can be maximized by choosing optimal values for x1 and x2. Express the given problem as linear programming problem in the standard form
2. State Duality Theorem.
3. Use the simplex method to find the maximum value of

   $$z = 2x_1 - x_2 + 2x_3$$

   subject to the constraints

   subject to the constraints

   $$\begin{aligned} 2x_1 + x_2 \quad\quad &\leq 10 \\ x_1 + 2x_2 - 2x_3 &\leq 20 \\ x_2 + 2x_3 &\leq 5 \end{aligned}$$

   where $x_1 \geq 0, x_2 \geq 0,$ and $x_3 \geq 0.$
4. Explain different ways to formulate the Problems as Linear Programs.
5. Formulate the given problem as linear programming problem: A prison is trying to decide what to feed its prisoners. They would like to offer some combination of milk, beans, and oranges. Their goal is to minimize cost, subject to meeting the minimum nutritional requirements imposed by law. The cost and nutritional content of each food, along with the minimum nutritional requirements are shown below.

	Milk (gallons)	Navy Beans (cups)	Oranges (large Calif. Valencia)	Minimum Daily Requirement
Niacin (mg)	3.2	4.9	0.8	13.0
Thiamin (mg)	1.12	1.3	0.19	1.5
Vitamin C (mg)	32.0	0.0	93.0	45.0
Cost ($)	2.00	0.20	0.25	

Course Outcome 4 (CO4):

1. Consider the k-server problem: The algorithm can move k servers in space; in the beginning they are placed on fixed points of a set M in space.
   Given is a sequence of requests _ = r1; r2; : : : ; rn, where each request corresponds to a point in the plane. A request ri is considered served, when a server has reached the point ri.
   The algorithm has to serve the requests in the given order by moving the servers. The cost of the algorithm is the sum of all distances that the servers have to move (according to some specified metric).
   Show that a greedy strategy for the k-server problem is not necessarily competitive. (Here, a greedy strategy chooses the cheapest possibility, i.e., it moves the server that is closest to the request.)
   Hint: Consider an example with k = 2 servers and an in_nite sequence on 3 well-chosen request points.
2. Given: k = 4 pages, numbered 1,2,3,4, and a cache of size 3. In the beginning, the pages 1,2,3 are residing in the cache. Consider the sequence δ = (41234) of requests.
   Wanted: For the marking algorithm, compare the strategies FIFO (first in - first out) and LFU (least frequently used).
   a. Which strategy generates more page faults for the sequence _ given above?
   b. How many page faults occur for FIFO and LFU, respectively?
   c. For each step, indicate which pages are residing in the cache.
3. For some problem P, two online algorithms A1 and A2 are given, with a competitive ratio of 2 and 3, respectively. Design a randomized online-algorithm with competitive ratio 9/4.
4. During the tutorials we presented the online algorithm SUM for the Bahncard Problem. This algorithm has a competitive ratio of 2 - β. Construct a sequence δ that reaches this competitive ratio (i.e., a worst-case example).
5. Prove the following statement:
   Let ALG be any marking algorithm as presented in the lecture with a cache of size k, and let OPT be an optimal offline-algorithm with a cache of size h ≤ k. Then ALG is k / (-h+1) competitive.
   Hint: Analogous to the problem in the lecture: consider a decomposition of a sequence δ into phases of length k.

Course Outcome 5 (CO5):

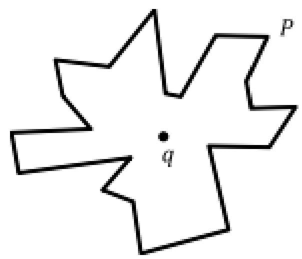
1. Given an undirected graph G = (V, E), a vertex cover of G is a subset V’ ⊆ V such that, for every edge (u,v) ⊆ E, we have either u ⊆ V’ or v ⊆ V’. Find the smallest possible vertex cover?
2. Explain set cover problem.
3. Show that Set Cover is NP-complete by making use of Vertex Cover.
4. Explain Vertex cover problem.
5. An instance of Set Cover is given by a ground set U = x1, x2,...,xn, a collection of m subsets Si ⊆ U of that ground set, and an integer k. Can you select a collection C of at most k of

these subsets such that taken together, they "cover" all of U? In other words, is there a 1 set C ⊆ {1, 2,...,m} such that |C| ≤ k and

$$\bigcup_{i \in C} S_i = U.$$

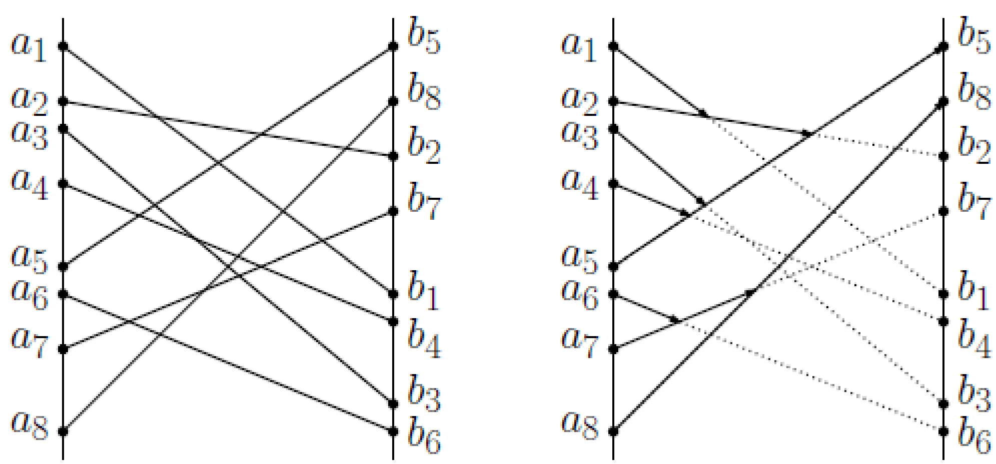
Course Outcome 6 (CO6):

1. Intersect the unit sphere in 3-space with a general quadratic surface. Show that the convex hull of the resulting curve C is a spectrahedron.
2. A polygon P is star-shaped if there exists a point q inside the polygon such that for every point p in the polygon P, the line pq is contained within P. You are given the polygon P as a counter clockwise list of n vertices (the position of q is not known). Give an O(n) algorithm to compute the convex hull of P (vertices listed in counter clockwise order). Prove the correctness of your algorithm.



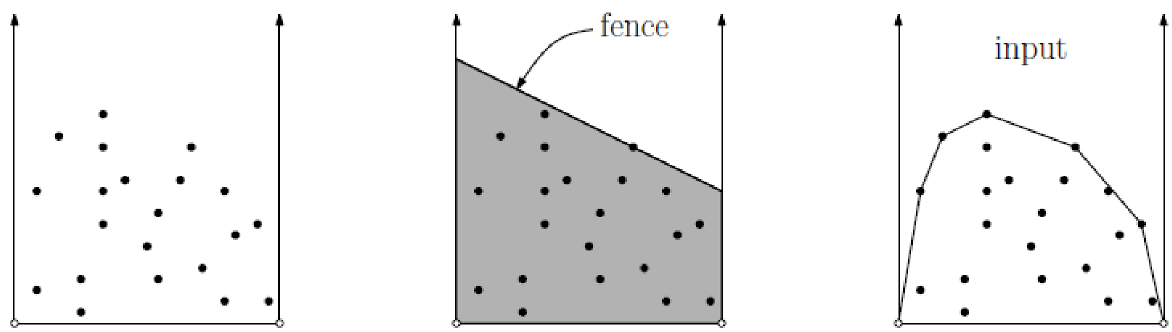
A star-shaped polygon

3. Draw the Voronoi diagram of 10 points all on a line. Draw separately the Voronoi diagram of 10 points all on a circle. What do these two diagrams have in common?
4. You are given two vertical lines and a set of n (nonvertical) line segments which pass between these vertical lines. Let ai denote the y-coordinate where the ith line segment hits the left vertical line and let bi denote the y-coordinate where it hits the right vertical line (see Fig. (a)). Scanning from left to right, whenever two segments intersect, the segment with the lower slope terminates and the one with the higher slope continues on (see Fig. (b)).
   Present an efficient algorithm that determines for each line segment which segment caused its termination. (For example, in the example shown in Fig. (b), segment 1 was terminated by segment 2, segment 2 was terminated by segment 5, segment 3 was terminated by segment 5, and so on.) If a segment was not terminated (as in the cases of segments 5 and 8), you can imagine that the right vertical segment is segment n + 1 of infinite slope that terminates all the segments that were not terminated before.

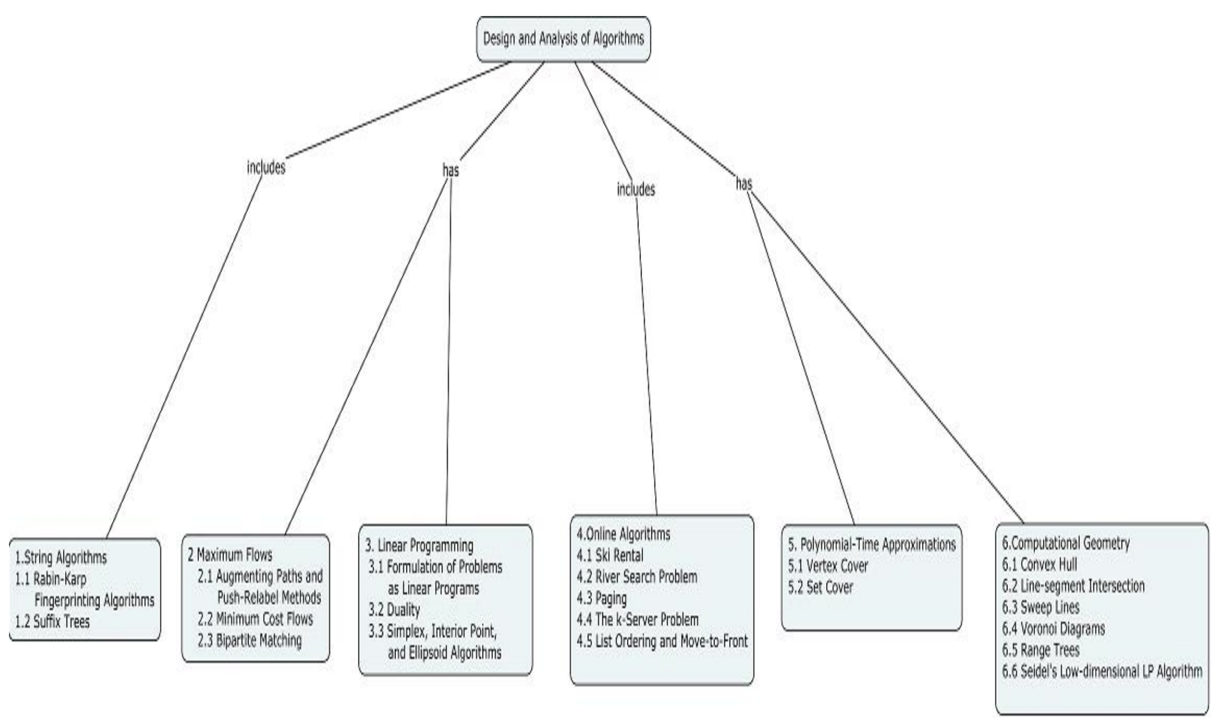


(a) (b)

5. A farmer has an orchard where various trees have been planted. Let P = {p1, . . . , pn} denote the coordinates of these trees. (The trees are very skinny, so each can be modeled by a single point.
   Maybe they are palm trees and the farmer sells coconuts.) The farm is bordered on east and west by roads running north-south and on the south by a road running east-west (see Figure below). The farmer wants to erect a straight-line fence to bound the north side of his orchard (to keep out those pesky coconut-eating armadillos). Since the city charges him tax based on the area of the farm, he wants to erect the fence to minimize the enclosed area (shaded in the figure).
   Present an efficient algorithm to determine where the fence should be placed. Assuming that the upper hull of the points has already been computed (including the lower left and right corners of the property), show that it is possible to determine where to put the fence in O(log n) time. (Hint: Begin by determining what geometric properties the area-minimizing line must satisfy. You will need to include a proof of this in your solution.)



Concept Map



Syllabus

String Algorithms - Rabin-Karp Fingerprinting Algorithm - Suffix Trees. **Maximum Flows** - Augmenting Paths and Push-Relabel Methods - Minimum Cost Flows - Bipartite Matching. **Linear Programming** - Formulation of Problems as Linear Programs – Duality - Simplex, Interior Point, and Ellipsoid Algorithms. **Online Algorithms** - Ski Rental - River Search Problem – Paging - The k-Server Problem - List Ordering and Move-to-Front. **Polynomial-Time Approximations** - Vertex Cover - Set Cover. **Computational Geometry** - Convex Hull - Line-segment Intersection - Sweep Lines - Voronoi Diagrams - Range Trees - Seidel's Low-dimensional LP Algorithm.

Text and Reference Books

1. J Kleinberg, E Tardos, Algorithm Design, Addison-Wesley, 2005.
2. TH Cormen, CF Leiserson, RL Rivest, C Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009.
3. AV Aho, J Hopcroft, JD Ullman, The Design and Analysis of Algorithms, Addison-Wesley, 1974.

Course Contents and Lecture Schedule

Module No.		Topic	No. of Lectures
1.		String Algorithms	
	1.1	Rabin-Karp Fingerprinting Algorithm	2
	1.2	Suffix Trees	2
2.		Maximum Flows	
	2.1	Augmenting Paths and Push-Relabel Methods	2
	2.2	Minimum Cost Flows	1
	2.3	Bipartite Matching	2
3.		Linear Programming	
	3.1	Formulation of Problems as Linear Programs	2
	3.2	Duality	1
	3.3	Simplex, Interior Point, and Ellipsoid Algorithms	2
4.		Online Algorithms	
	4.1	Ski Rental	2
	4.2	River Search Problem	1
	4.3	Paging	2
	4.4	The k-Server Problem	1
	4.5	List Ordering and Move-to-Front	2
5.		Polynomial-Time Approximations	

Module No.		Topic	No. of Lectures
	5.1	Vertex Cover	2
	5.2	Set Cover	2
6.		Computational Geometry	
	6.1	Convex Hull	2
	6.2	Line-segment Intersection	1
	6.3	Sweep Lines	2
	6.4	Voronoi Diagrams	1
	6.5	Range Trees	2
	6.6	Seidel's Low-dimensional LP Algorithm	2

Course Designer:

1. Dr.M.K.KavithaDevi mkkdit@tce.edu

		Category	L	T	P	Credit
14CSPK0	**REAL TIME OPERATING SYSTEMS**	PE	3	0	0	3

Preamble

The student will be able to understand the concepts of Real-Time operating systems and Scheduling of Real-Time algorithm. This course also includes set of case studies that provides insight into some existing operating systems.

Prerequisite

14CS420 – System Software and Operating Systems

Course Outcomes

On the successful completion of the course, students will be able to

Explain the concepts, Characteristics and issues of Real-Time systems. (CO1)	Understand
Apply the concepts of different Real-Time Scheduling algorithm, in Real-Time operating system environment. (CO2)	Apply
Implement the concepts of buffering, mailbox, semaphore, critical section and dead lock avoidance in Real-Time Operating Systems.(CO3)	Apply
Explain the features of Real-Time Operating System as Commercial Real-Time Operating Systems. (CO4)	Understand
Identify the role of Real-Time operating system in Contemporary Real-Time Operating Systems (CO5)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M											
CO2.	M	M	M	M								
CO3.	M	M	M	M								
CO4.	M	M	L	M							L	
CO5.	M	M	M	M	L						L	

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	30	30	20

Understand	30	40	40	40
Apply	40	30	30	40
Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define Real Time System.
2. List out the some typical real time domains and applications.
3. Discuss about the different Real-Time system design Issues.
4. Explain the different characteristics of Real-Time Systems.

Course Outcome 2 (CO2):

1. Define Context switching.
2. What do you understand by Scheduling point of a task scheduling algorithm?
3. Discuss some of the advantages of EDF scheduling over RM Scheduling and vice versa.
4. Consider a real time system which consists of three tasks $T_{1,}T_2$ and T_3 which have been Characterized in the following table.

Task	Phase(mSec)	Execution Time(m Sec)	Relative Deadline(m Sec)	Period(m Sec)
T_1	20	10	20	20
T_2	40	10	50	50
T_3	70	20	80	80

If the tasks are to be scheduled using a table-driven scheduler, what is the length of time for which the schedules have to be stored in a precomputed schedule table of the scheduler.

Course Outcome 3 (CO3)

1. What effect would size N of ring buffer have on its performance? How would you determine the optimal size?
2. Discuss in detail about Banker's algorithm for deadlock avodence.
3. Illustrate the Priority Inheritance Protocol.

Course Outcome 4 (CO4)

1. As the developer of hard real-time applications, explain the features that you consider necessary for Real-Time Operating System(RTOS) to support.

2. Explain the time service that a RTOS is expected to support. Also briefly highlight how timer service are implemented in a real-time operating system .

3. Discuss in detail about features of Real-Time Operating system.

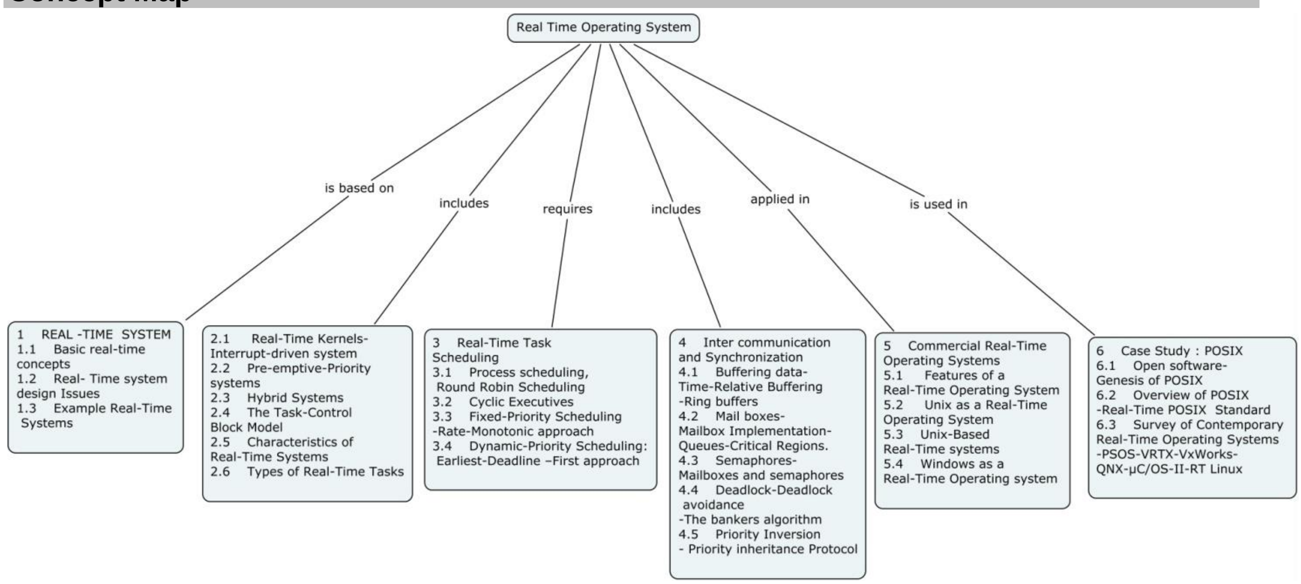
Course Outcome 5 (CO5)

1. List four important features that a POSIX 1003.4(Real-Time standard) compliant operating system must support.

2. Explain how interrupts are handled in windows NT. Explain how interrupt processing scheme of windows NT makes it unsuitable for hard real-time applications.

3. Identify at least four important advantages of using Vx works as the operating system for large hard real-time applications compared to using Unix V.3

Concept Map



Syllabus

Basic real-time concepts- Real- Time system design Issues-Example Real-Time Systems- **Real-Time Operating systems**-Real-Time Kernels- Pre-emptive-Priority systems- Hybrid Systems- The Task-Control Block Model- Characteristics of Real-Time Systems- Types of Real-Time Tasks. **Real-Time Task Scheduling-**Process scheduling, Round Robin Scheduling-Cyclic Executives- Fixed-Priority Scheduling-Rate-Monotonic approach- Dynamic-Priority Scheduling: Earliest-Deadline –First approach- **Inter communication and Synchronization-** Buffering data-Time-Relative Buffering-Ring buffers- Mail boxes-Mailbox Implementation-Queues-Critical Regions.- Semaphores-Mailboxes and semaphores- Deadlock-Deadlock avoidance-The bankers algorithm-Priority Inversion- Priority inheritance Protocol- **Commercial Real-Time Operating Systems-** Features of a Real-Time Operating System- Unix as a Real-Time Operating System- Unix-Based Real-Time systems- Windows as a Real-Time Operating system- **Case Study :- POSIX-** Open software-Genesis of POSIX- Overview of POSIX-Real-Time POSIX Standard- Survey of Contemporary Real-Time Operating Systems-PSOS-VRTX-VxWorks-QNX-µC/OS-II-RT Linux.

Text Books:

1. Philip A. Laplante, "Real time systems design and analysis" –,Third Edition.IEEE Press-Wiley India-Copyright 2005 and reprint -2010.
2. Rajib Mall –"Real-Time Systems" Theory and Practice-copy right 2007-Pearson Education.

Reference Books

1. Allan. V. Shaw, "Real Time systems and software", John Wiley & Sons(4,5)
2. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Prentice Hall of India - Indian Edition – 1995.
3. Wayne Wolf, "Computers as Components: Principles of Embedded Computing System Design", Elsevier Inc ,2001,(4,5)
4. C.M. Krishna, Kang G. Shin, "Real–Time Systems", McGraw – Hill International Editions, 1997.
5. Raymond J.A.Bhur, Donald L.Bailey, "An Introduction to Real Time Systems", PHI 1999
6. R.J.A Buhur, D.L Bailey, "An Introduction to Real – Time Systems", Prentice – Hall International, 1999.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Real -Time System (4)**	
1.1	Basic real-time concepts	1
1.2	Real- Time system design Issues	2
1.3	Example Real-Time Systems	1
2	**Real-Time Operating Systems(8)**	
2.1	Real-Time Kernels - Interrupt-driven system	1
2.2	Pre-emptive-Priority systems	1
2.3	Hybrid Systems	2
2.4	The Task-Control Block Model	2
2.5	Characteristics of Real-Time Systems	1
2.6	Types of Real-Time Tasks	1
3	**Real-Time Task Scheduling (6)**	
3.1	Process scheduling, Round Robin Scheduling	1
3.2	Cyclic Executives	1

Module No.	Topic	No. of Lectures
3.3	Fixed-Priority Scheduling-Rate-Monotonic approach	2
3.4	Dynamic-Priority Scheduling: Earliest-Deadline –First approach	2
4	**Inter communication and Synchronization(8)**	
4.1	Buffering data-Time-Relative Buffering-Ring buffers	1
4.2	Mail boxes-Mailbox Implementation-Queues-Critical Regions.	2
4.3	Semaphores-Mailboxes and semaphores	1
4.4	Deadlock-Deadlock avoidance-The bankers algorithm	2
4.5	Priority Inversion- Priority inheritance Protocol	2
5	**Commercial Real-Time Operating Systems (7)**	
5.1	Features of a Real-Time Operating System	1
5.2	Unix as a Real-Time Operating System	2
5.3	Unix-Based Real-Time systems	2
5.4	Windows as a Real-Time Operating system	2
6	**Case Study : POSIX(4)**	
6.1	Open software-Genesis of POSIX	1
6.2	Overview of POSIX-Real-Time POSIX Standard	1
6.3	Survey of Contemporary Real-Time Operating Systems-PSOS-VRTX-VxWorks-QNX-µC/OS-II-RT Linux	1
	Total	**36**

Course Designer:

1. Mr.R.Chellamani rcmcse@tce.edu

		Category	L	T	P	Credit
14CSPL0	**USER INTERFACE DESIGN**	PE	3	0	0	3

Preamble

This course is designed to be undertaken by those who need to apply the knowledge of the role of human factors/ergonomics in the design, evaluation and use of interactive systems.
It will provide an understanding of the key issues related to the design, evaluation and use of user centered technologies and provide you with additional resources to facilitate the learning process.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Outline the importance of the user interface in software development.(CO1)	Understand
List the key aspects of human psychology which can determine user actions at and satisfaction of the interface.(CO2)	Remember
Describe the key design principles for user interfaces.(CO3)	Create
Set up and carry out a process to gather requirements for, engage in iterative design of, and evaluate the usability of a user interface.(CO4)	Analyze
Constuct how user interface development can be integrated into an overall software development process.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S											
CO2.		S		M								
CO3			S	M								
CO4	S				S							
CO5				S							S	

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	10	10	10	20
Understand	10	10	10	20
Apply	10	10	10	20

Analyse	20	10	0	20
Create	0	10	20	20

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Discuss the characteristics of graphical interface.
2. Define configurability.
3. List any 4 difference between GUI versus web page design.
4. What are the advantages of cascading menu?
5. Define interactive paper prototype.

Course Outcome 2 (CO2):

1. List any four principles of UID.
2. What is mean by visually pleasing composition?
3. What is a seizure disorder?
4. What is hypermedia?
5. Give the rules to develop layout grids.

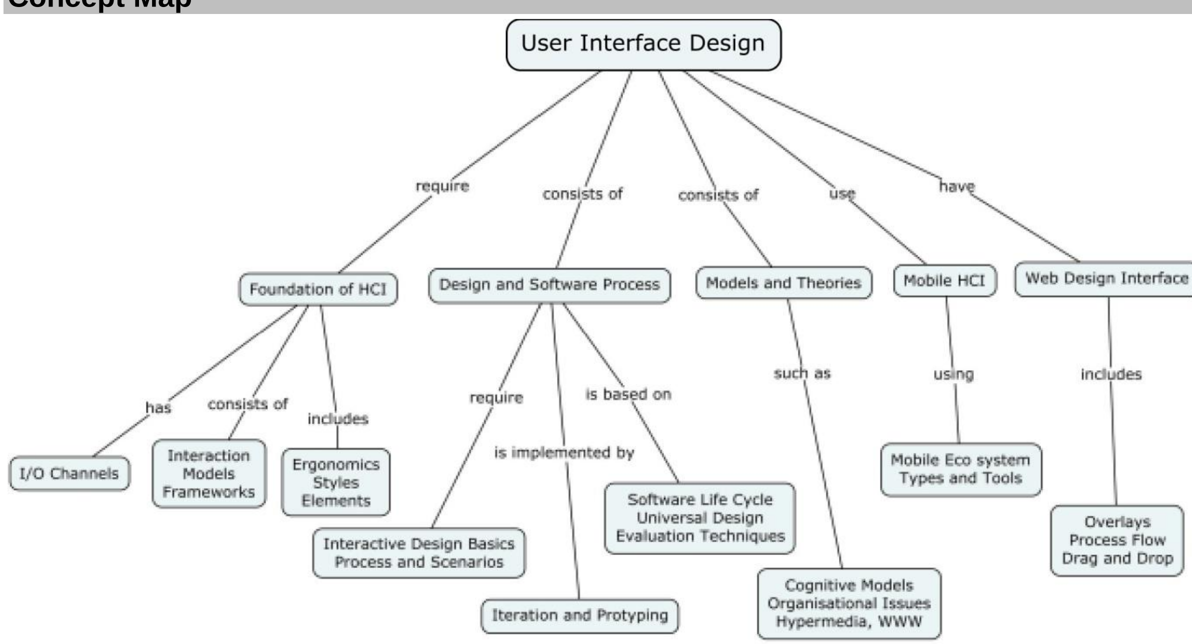
Course Outcome 3 (CO3):

1. Elaborate the general principles of User Interface Design.
2. Design Icons for an application.
3. Explain www with suitable example.

Course Outcome 4 (CO4):

1. Compare and Contrast between GUI and Web Interface.
2. Explain in detail the functions of menus and formatting of menus.
3. Analyze in detail about components and presentation styles of windows with suitable illustrations.

Concept Map



Syllabus

FOUNDATIONS OF HCI : The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.
DESIGN & SOFTWARE PROCESS: Interactive Design basics – process – scenarios – navigation – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice –. Design rules –.Evaluation Techniques – Universal Design.
MODELS AND THEORIES: Cognitive models – Organizational issues and stake holder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.
MOBILE HCI: Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Mobile Design: Elements of Mobile Design, Tools.
WEB INTERFACE DESIGN: Designing Web Interfaces – Drag & Drop, Contextual Tools, Overlays, Process Flow. Case Studies.

Text Book

1. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human Computer Interaction", 3rd Edition, Pearson, 2004.

Reference Books

1. Brian Fling, "Mobile Design and Development", First Edition , O‟Reilly Media Inc., 2009
2. Bill Scott and Theresa Neil, "Designing Web Interfaces", First Edition, O‟Reilly, 2009

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Foundations of HCI**	
1.1	The Human: I/O channels	1
1.2	Reasoning and problem solving	1
1.3	The computer: Devices ,Memory, processing and networks	1
1.4	Interaction: Models :Frameworks	1
1.5	Ergonomics	1
1.6	Styles, Elements	1
1.7	Interactivity, Paradigms	2
2.	**Design & Software Process**	
2.1	Interactive Design basics	1
2.2	Process, Scenarios and navigation	1
2.3	Iteration and prototyping	1
2.4	HCI in software process, Software life cycle	2
2.5	Usability engineering, Prototyping in practice	1
2.6	Design rules, Evaluation Techniques	2
2.7	Universal Design.	1
3.	**Models And Theories**	
3.1	Cognitive models	2
3.2	Organizational issues and stake holder requirements	1
3.3	Communication and collaboration models	2
3.3	Hypertext, Multimedia and WWW	2
4.	**Mobile HCI**	
4.1	Mobile Ecosystem: Platforms, Application frameworks	2

Module No.	Topic	No. of Lectures
4.2	Types of Mobile Applications: Widgets, Applications,	2
4.3	Mobile Design: Elements of Mobile Design, Tools	2
5.	**Web Interface Design**	
5.1	Designing Web Interfaces, Drag & Drop	2
5.2	Contextual Tools, Overlays	2
5.3	Process Flow. Case Studies.	2

Course Designer:

1. Mr. V.Vignaraj Ananth vignaraj@tce.edu

14CSPM0	CLOUD COMPUTING AND VIRTUALIZATION	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course is offered as an elective to the Under Graduate students of Computer Science and Engineering. This course is aimed at introducing cloud computing, the services offered by the cloud, Virtualization, Cloud Storage and Cloud Security.

Prerequisite

- 14CS370 - Object Oriented Programming
- 14CS420 - System Software and Operating Systems
- 14CS520 - Computer Networks
- 14CS620 - Internet programming

Course Outcomes

On the successful completion of the course, the students will be able to

Discuss the advantages and disadvantages of the cloud paradigm and compare the cloud service models.(CO1)	Understand
Illustrate the design of a datacenter and discuss about IT service management.(CO2)	Apply
Identify the various forms of virtualization techniques that can be applied to multiple layers within a datacenter.(CO3)	Apply
Illustrate the cloud security mechanisms that can be applied to combat the various cloud security threats.(CO4)	Apply
Illustrate the data availability, data replication, data protection and data footprint reduction techniques of cloud storage services.(CO5)	Understand
Illustrate the capabilities of Microsoft Azure cloud and OpenStack.(CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L				L	L				
CO2	L	L	L		L							
CO3	S	M	M		S		M	L	L			L
CO4	S	M	M		S		M	M	L			L
CO5	S	L	L		S		M	L	L			L
CO6	S	M	M		S		M	M	L			L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			End-semester examination
	1	2	3	
Remember	30	20	30	20
Understand	30	30	30	40
Apply	40	50	40	40

Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1)

1. Discuss the layers and types of cloud. (Understand)
2. Discuss on the IaaS cloud service model. (Understand)
3. Explain the desired features of a Cloud. (Understand)

Course Outcome 2 (CO2)

1. Discuss on Cloud Orchestration. (Understand)
2. Explain the design considerations that need to looked upon while designing a datacenter. (Understand)
3. Discuss on IT service management. (Understand)

Course Outcome 3 (CO3)

1. Demonstrate the type of virtualization that is supported by the virtualization tool named 'PlateSpin Power Recon'. (Apply)
2. Illustrate the steps to add the OpenSolaris Guest OS to Sun xVM VirtualBox. (Apply)
3. Define Internal Network Virtualization. (Remember)

Course Outcome 4 (CO4)

1. Explain the security threats that can be anticipated in a cloud environment.(Understand)
2. Explain the use of cloud-based security groups that are used to counter and prevent the cloud security threats and attacks. (Understand)
3. Experiment with symmetric encryption mechanism to illustrate the prevention of security threats and attacks with the help of an example. (Apply)

Course Outcome 5 (CO5)

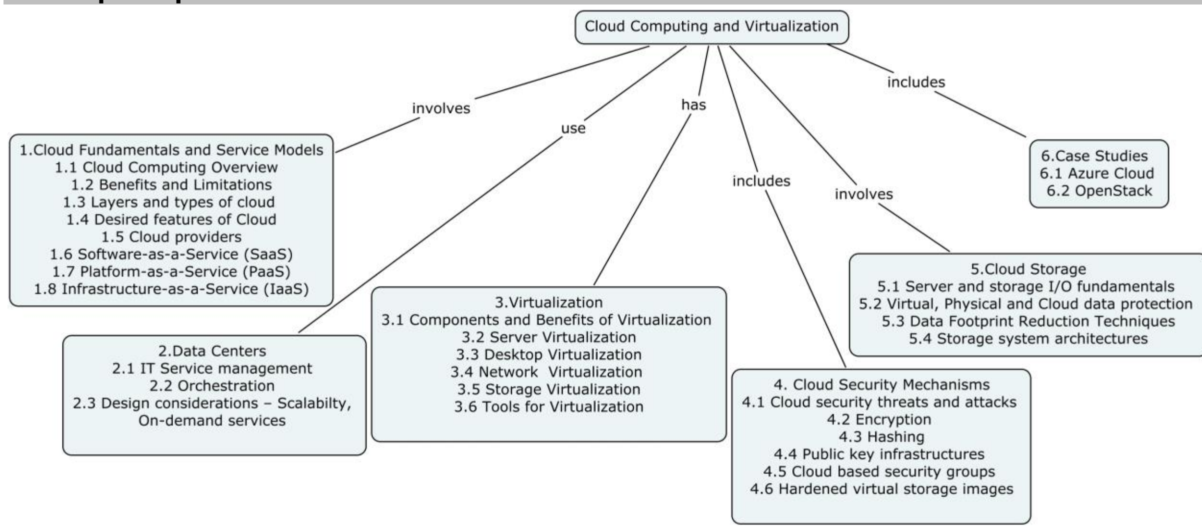
1. Explain how the DFR techniques can be applied to your cloud to provide capacity optimization. (Understand)
2. Discuss on the types of disk storage. (Understand)
3. Define a space-saving snapshot. (Remember)

Course Outcome 6 (CO6)

1. Discuss on the steps to create a virtual machine with an already installed Java Development Kit (JDK). (Understand)

2. Write a script using PowerShell to balance the load across the Virtual machines using PowerShell in Azure. (Apply)
3. Illustrate how a computational resource can be orchestrated using the key components in OpenStack. (Apply)

Concept Map



Syllabus

Cloud Fundamentals and Service Models - Cloud Computing Overview – benefits – limitations – Layers and types of cloud – Desired features of Cloud – Cloud providers - IT Services Delivery Model - Software-as-a-Service (SaaS) - Platform-as-a-Service (PaaS) - Infrastructure-as-a-Service (IaaS) – **Data Centers** – IT Service management – Orchestration – Design considerations – Scalabilty, On-demand services - **Virtualization –** Components and Benefits of Virtualization - Server Virtualization - Application Virtualization - Desktop Virtualization - Network Virtualization – Storage virtualization - Tools for Virtualization – **Cloud Security Mechanisms –** Cloud security threats and attacks – Encryption – Hashing – Public key infrastructures – Cloud based security groups – Hardened virtual storage images – **Cloud Storage –** Server and storage I/O fundamentals – Virtual, Physical and Cloud data protection - Data Footprint Reduction Techniques – Storage system architectures – **Case Studies** – Azure Cloud – OpenStack

Reference Books

1. Thomas Erl, Zaigham Mahmood, Ricardo Puttini, “Cloud Computing: Concepts, Technology & Architecture”, Prentice Hall Service Technology Series, 2013.
2. John Rittinghouse, James Ransome, “Cloud Computing: Implementation, Management and Security”, CRC Press 2010.

3. Greg Schulz, “Cloud and Virtual Data Storage Networking”, CRC Press, 2012.
4. Nelson Ruest, Danielle Ruest, “Virtualization, A Beginner’s Guide”, McGraw-Hill Companies, 2009.
5. http://www-935.ibm.com/services/us/en/it-services/it-service-management-implementation.html
6. www.bcs.org/upload/pdf/itsm-guide-foundation.pdf
7. https://www.vmware.com/files/pdf/analysts/IDC-White-Paper_MGMT.pdf

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Cloud Fundamentals and Service Models (8)**	
1.1	Cloud Computing Overview	1
1.2	Benefits and Limitations	1
1.3	Layers and types of cloud	1
1.4	Desired features of Cloud	1
1.5	Cloud providers	1
1.6	IT Services Delivery Model - Software-as-a-Service (SaaS)	1
1.7	Platform-as-a-Service (PaaS)	1
1.8	Infrastructure-as-a-Service (IaaS)	1
2	**Data Centers (4)**	
2.1	IT Service management	1
2.2	Orchestration	1
2.3	Design considerations – Scalability, On-demand services	2
3	**Virtualization (7)**	
3.1	Components and Benefits of Virtualization	1
3.2	Server Virtualization	2
3.3	Desktop Virtualization	1
3.4	Network Virtualization	1
3.5	Storage Virtualization	1
3.6	Tools for Virtualization	1

4	**Cloud Security Mechanisms (6)**	
4.1	Cloud security threats and attacks	1
4.2	Encryption	1
4.3	Hashing	1
4.4	Public key infrastructures	1
4.5	Cloud based security groups	1
4.6	Hardened virtual storage images	1
5	**Cloud Storage (5)**	
5.1	Server and storage I/O fundamentals	1
5.2	Virtual, Physical and Cloud data protection	1
5.3	Data Footprint Reduction Techniques	2
5.4	Storage system architectures	1
6	**Case Studies (6)**	
6.1	Azure Cloud	3
6.2	OpenStack	3

Course Designer:

1. Mrs.J.Jane Rubel Angelina janerubel@tce.edu

14CSPN0	INTERNET OF THINGS AND ITS APPLICATIONS	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course aims at providing a basic understanding of Internet of Things, exemplifying the application areas where Internet of Things can be applied and enables designing prototypes of Internet-connected products using appropriate tools.

Prerequisite

14CS521: Computer Networks

14CS270: Problem Solving Using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Appreciate the significance of IoT,WoT and Cloud of Things (CO1)	Understand
Describe the general IoT architecture and connected domains. (CO2)	Understand
Analyze the requirement to figure out the suitable communication technology and protocols required for an IOT application.(CO3)	Analyze
Explain the challenges in wearable computing, components of wearable technology and types of wearable.(CO4)	Understand
Develop an IoT application for the given specification applying the IoT technologies.(CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L											
CO2	L											
CO3	S	M	M	L		L	M					L
CO4	M				L		L					L
CO5	S	M	S		L	L	M	L				L

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	50	30	20	20
Understand	40	40	30	30
Apply	10	10	40	40
Analyse	0	20	10	10
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate On premises and connected system.
2. Define Web of Things.
3. List the IoT levels.
4. Discuss the two pillars of WoT.
5. Refine the role of data science in connected environment.
6. Define various IoT implementation platforms and boards.
7. List the limitations in on premises technology.

Course Outcome 2 (CO2):

1. Illustrate the need of Gateways/IoT Routers in connected environment?
2. Explain the connected home concept and the sub systems involved in that.
3. Define Integrated Security and indicate the role of IoT in that.
4. Summarize the work flow of centralized alarm management through IoT End Points.
5. Describe automation hubs and bridges.

Course Outcome 3 (CO3):

1. Compare the conventional network protocols characteristics with IoT protocols and analyze the need for IoT protocols.
2. Propose a design for Secure Development of IoT applications.
3. Examine the pros and cons of MQTT, CoAP as IoT protocols for resource-constrained devices?
4. Survey the pros and cons of the request-respond model versus publish-subscribe model in the context of web protocols used in Internet of Things?
5. Find the communication technology mediums that can be over clocked for increased performance.

Course Outcome 4 (CO4):

1. List the types of wearables.
2. Indicate the advantages of implementing edge analytics on wearable devices.
3. Define virtual contimum.
4. Differentiate augment reality, virtual reality, augment virtuality and mixed reality.
5. Identify the raw parameters that can be refined from end user for analytics using smart watch.

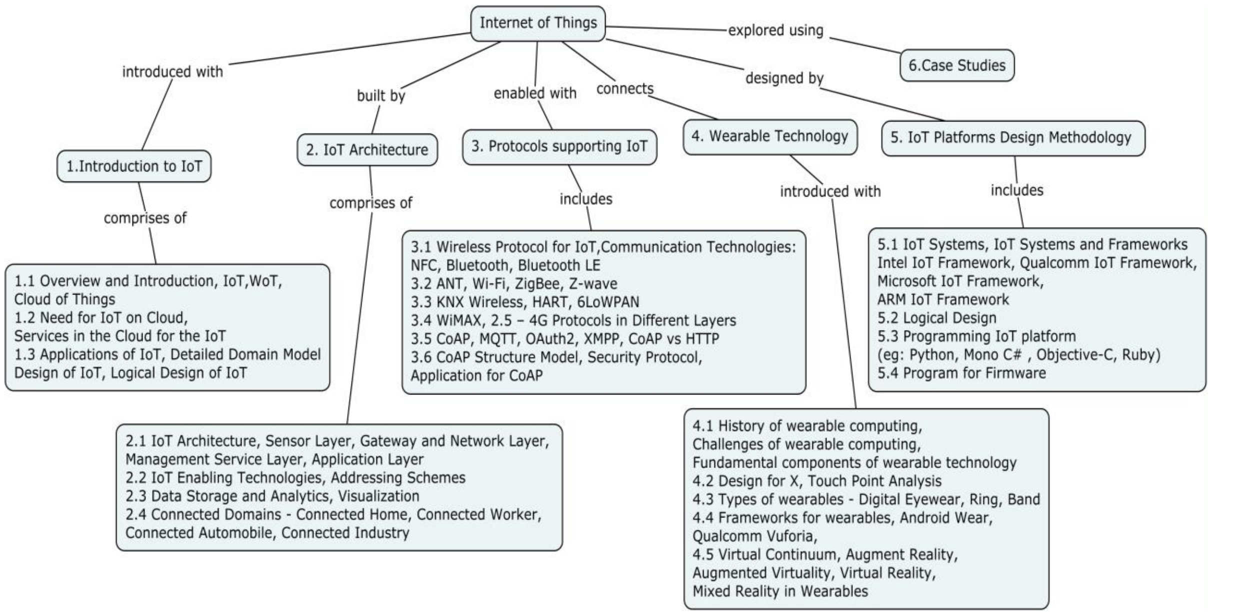
Course Outcome 5 (CO5):

1. What are the architectural risks associated while using radio for near field communication in the IoT applications.
2. Propose an alternate design for using 801.2 Wi-Fi communications for distributed communication over the sensor nodes.
3. Demonstrate the IoT design methodology for smart irrigation.
4. For an existing street light management system, do a Design for X to scale them to 1 to n groups.
5. Write a program to read values from the sensors and print their values on the console. The printed values should be a conversion of the ADC values to their human readable counterparts.
6. Write a the blink program so that each LED blinks with its own independent schedule of 0.25 Hz for the Red LED, 0.5 Hz for the Green LED and 1 Hz for the Blue LED.
7. Write a program for light sensing device. The light intensity should be classified into dark, light, medium, and strong light. Three LEDs should be used to indicate whether it is

dark (no LED turned on), there is low light (one LED turned on), medium light (two LEDs turned on) or strong light (all LED turned on).

8. Write a program to make LED blink. LED blinks with its own independent schedule of 0.25 Hz for the Red LED, 0.5 Hz for the Green LED and 1 Hz for the Blue LED.

Concept Map



Syllabus

Introduction to IoT: Overview and Introduction, Internet of Things (IoT), Web of Things (WoT), Cloud of Things, Need for IoT on Cloud, Services in the Cloud for the Internet of Things, Applications of IoT – Detailed Domain Model. **IoT Architecture:** IoT Architecture, Sensor Layer, Gateway and Network Layer, Management Service Layer, Application Layer, IoT Enabling Technologies, Addressing Schemes, Data Storage and Analytics, Visualization. Connected Domains – Connected Home, Connected Worker, Connected Automobile, Connected Industry. **Protocols Supporting IoT:** Wireless Protocol for IoT, Communication Technologies - NFC, Bluetooth, Bluetooth LE, ANT, Wi-Fi, ZigBee, Z-wave, KNX Wireless, HART, 6LoWPAN, WiMAX, 2.5–4G Protocols in Different Layers, Architecture, Features & Functions of CoAP, MQTT, OAuth2, XMPP, CoAP vs HTTP, CoAP Structure Model, Security Protocol and Application for CoAP. **Wearable Technology:** History of wearable computing, Challenges of wearable computing, Fundamental components of wearable technology, Design for Excellence, Touch Point Analysis, Types of Wearables - Digital Eyewear, Ring, Band, Frameworks for wearable, Android Wear, Qualcomm Vuforia, Virtual Continuum, Augment Reality, Augmented Virtuality, Virtual Reality, Mixed Reality in Wearables. **IoT Platforms Design Methodology :** IoT Systems – Intel IoT Framework, Qualcomm IoT Framework, Microsoft IoT Framework, ARM IoT Framework, Logical Design, Programming IoT platform (eg: Python, Mono C# , Objective-C, Ruby), Program for Firmware – **Case Studies**

Text Books

1. Olivier Hersent, David Boswarthick and Omar Elloumi, The Internet of Things: Key Applications and Protocols, Second Edition, Wiley Publisher, 2012
2. Uckelmann, Dieter, Mark Harrison, and Florian Michahelles, Architecting the Internet of Things. Springer Science & Business Media, 2011.

3. Jean-Philippe Vasseur, Adam Dunkels, Interconnecting Smart Objects with IP: The Next Internet, Morgan Kuffmann, 2010
4. Jonathan L. Zittrain, The Future of the Internet, Yale University Press & Penguin UK 2008.
5. Samuel Greengard, The Internet of Things The Internet of Things (The MIT Press Essential Knowledge series), MIT Press, 2015

Reference Books

1. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands on Approach, 2014
2. Doukas, Charalampos, Building internet of things with the Arduino, CreateSpace Independent Publishing Platform, 2012.
3. Lu, Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, CRC Press
4. Massimo Banzi, Getting Started with Arduino (Make: Projects). O'Reilly Media. 2008.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Introduction to IoT**	
1.1	Overview and Introduction, Internet of Things (IoT), Web of Things (WoT), Cloud of Things	1
1.2	Need for IoT on Cloud, Services in the Cloud for the Internet of Things	1
1.3	Applications of IoT, Detailed Domain Model	1
2.	**IoT Architecture**	
2.1	IoT Architecture, Sensor Layer, Gateway and Network Layer, Management Service Layer, Application Layer	1
2.2	IoT Enabling Technologies, Addressing Schemes	1
2.3	Data Storage and Analytics, Visualization	1
2.4	Connected Domains - Connected Home, Connected Worker, Connected Automobile, Connected Industry	1
3	**Protocols supporting IoT**	
3.1	Wireless Protocol for IoT, Communication Technologies - NFC, Bluetooth, Bluetooth LE	1
3.2	ANT, Wi-Fi, ZigBee, Z-wave	2
3.3	KNX Wireless, HART, 6LoWPAN	1
3.4	WiMAX, 2.5 – 4G Protocols in Different Layers	2
3.5	CoAP, MQTT, OAuth2, XMPP, CoAP vs HTTP	2
3.6	CoAP Structure Model, Security Protocol & Application for CoAP	2
4	**Wearable Technology**	
4.1	History of wearable computing, Challenges of wearable computing, Fundamental components of wearable technology	1
4.2	Design for X, Touch Point Analysis	2
4.3	Types of wearables - Digital Eyewear, Ring, Band,	2
4.4	Frameworks for wearables, Android Wear, Qualcomm Vuforia,	1
4.5	Virtual Continuum, Augment Reality, Augmented Virtuality, Virtual Reality, Mixed Reality in Wearables	1
5	**IoT Platforms Design Methodology :**	
5.1	IoT Systems and Frameworks – Intel IoT Framework,	1

Module No.	Topic	No. of Lectures
	Qualcomm IoT Framework, Microsoft IoT Framework, ARM IoT Framework	
5.2	Logical Design	2
5.3	Programming IoT platform (eg: Python, Mono C# , Objective-C, Ruby)	4
5.4	Program for Firmware	3
6	**Case Studies**	2
	Total	**36**

Course Designer:

1. M.Vijayalakshmi mviji@tce.edu

14CSPP0	BIG DATA ANALYTICS	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course aims at facilitating the student to explore and understand the Big data platform, its architecture and its technology foundations. Work on hadoop platform. Perform mining and analysis on massive data using certain techniques. Also perform analysis through visualization techniques.

Prerequisite

14CS440 – Database Management Systems

14CSPBO – Data Warehousing and Mining

Course Outcomes

On the successful completion of the course, students will be able to

Explain the big data perspective and its real world requirement (CO1)	Understand
Illustrate the working principle of big data architecture and its technology foundations. (CO2)	Understand
Compare and contrast the nature of data in distributed file systems (CO3)	Understand
Utilize the hadoop platform to work on huge data. (CO4)	Apply
Make use of certain analytical techniques on big data. (CO5)	Apply
Determine the results of big data analysis using certain analytical techniques or tools. (CO6)	Evaluate

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S											
CO2.	S											
CO3.	S											
CO4.	S	M	M	L					L			
CO5.	S	M	M	L	M				L			
CO6.	S	S		L	M				L			

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	40	20	20	10
Understand	60	30	30	30
Apply	0	50	50	60
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):
1. Define Big data.
2. What is the need for big data in business? List two examples.
3. How to manage big data?

Course Outcome 2 (CO2):
1. Explain the architecture of Big data with a neat sketch.
2. Define Virtualization.
3. Outline the role of cloud in big data.

Course Outcome 3 (CO3) :
1. Compare and contrast NoSQL with Relational databases.
2. Summarize the difference of data in warehouse and hadoop environment.
3. List the importance of warehouse in big data.

Course Outcome 4 (CO4) :
1. Make use of hadoop framework for explaining the role of it in big data analytics.
2. What do you mean by map reduce?
3. Utilizing an application explain the use of hadoop framework to maintain big data.

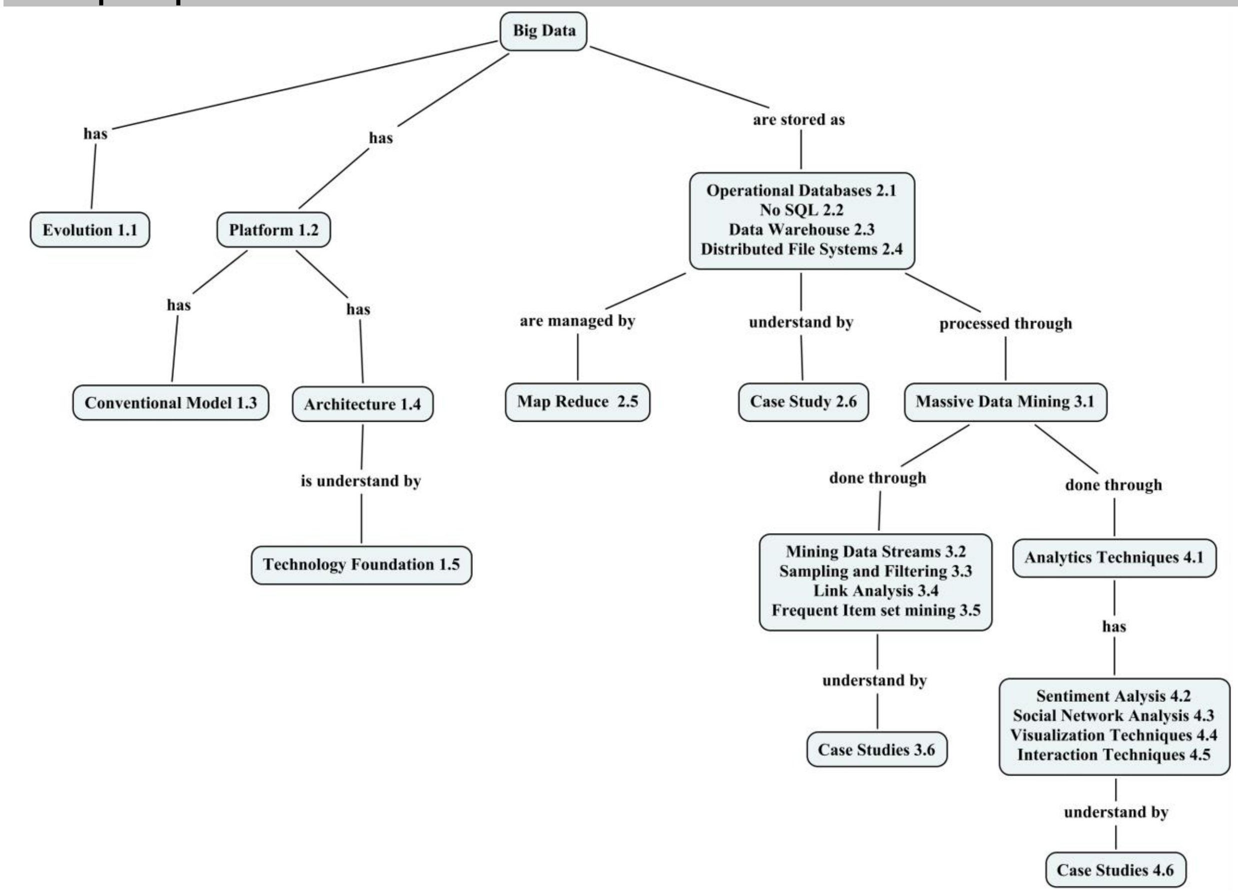
Course Outcome 5 (CO5) :
1. Compare and contrast sampling and filtering.
2. Define sentiment analysis.
3. What do you mean by stop words? Give examples.
4. Outline the interaction techniques.

5. Apply the association technique for the given data set to identifying frequent item sets.

Course Outcome 6 (CO6) :

1. Big data analytics in real world applications – The results can be analyzed and evaluated using tools. (Assignment / Mini Project)

Concept Map



Syllabus

Big Data Introduction: Evolution of data management, Big data definition, Big data a business perspective, Introduction to Big Data Platform – Challenges of Conventional Systems, How to manage big data, Building a successful architecture, Technology foundations of big data. **Big Data Management**: Operational databases, NoSQL, Distributed file systems and its architecture; Data Warehouse vs Hadoop, Map reduce fundamentals, Case study: Hadoop, HDFS, MongoDB. **Massive Data Mining:** Mining data streams – Model, Sampling, Filtering, and Counting Distinct Elements in a Stream, Link Analysis, and Frequent Itemset mining from data stream, Case Studies - Real World Analysis, Stock Market Predictions. **Analytics on big Data:** Defining big data analytics, Analytics techniques – Sentiment analysis, Social Network analysis, Understanding text data analytics, Visualizations - Visual data analysis techniques, Interaction techniques, Case study: Big data analytics in Health care systems.

Reference Books

1. Judith Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman, “Big Data for Dummies”, Wiley Brand, 2013.
2. Paul Zikopoulos, Chris Eaton, ”Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”, McGraw-Hill Osborne Media, 2011.
3. Anand Rajaraman, Jeffrey D. Ullman, “Mining of Massive Datasets”, Cambridge University Press New York, 2011.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Big data introduction (6)**	
1.1	Evolution of data management	1
1.2	Introduction to Big Data Platform	1
1.3	Challenges of Conventional Systems	1
1.4	Building a successful big data architecture	2
1.5	Technology foundations of big data	1
2	**Big Data Management (13)**	
2.1	Operational databases	1
2.2	NoSQL - MongoDB	2
2.3	Data in Warehouse	1
2.4	Data in Distributed File Systems – Hadoop, HDFS architectures	4
2.5	Map reduce fundamentals	3
2.6	Case study: Applications on Big Data Using Pig and Hive	2
3	**Massive Data Mining (10)**	
3.1	Massive data mining - introduction	1
3.2	Mining data streams	2
3.3	Sampling, Filtering, and Counting Distinct Elements in a Stream	1
3.4	Link Analysis	2
3.5	Frequent Item set mining from data stream	2
3.6	Case Studies - Real world analysis - Stock Market Predictions	2
4	**Analytics on big Data (7)**	
4.1	Big data analytic definition and techniques	1

Module No.	Topic	No. of Lectures
4.2	Sentiment Analysis	1
4.3	Social network analysis	1
4.4	Visual data analysis	1
4.5	Interaction techniques	1
4.6	Case study: Big data analytics in Health care systems	2
	Total	**36**

Course Designers:

1. Mrs. A.M.Rajeswari amrcse@tce.edu

14CSPQ0	**SOFT COMPUTING TECHNIQUES**	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

Soft Computing is a collection of computational techniques in computer science, artificial intelligence and engineering disciplines which attempt to study, model and analyze complex problems for which conventional methods were not able to produce low cost and complete solutions. Unlike conventional computing, soft computing techniques are tolerant of imprecision, uncertainty and approximations. A typical course in soft computing generally comprises of computational techniques like Genetic/Evolutionary algorithms, Artificial Neural Networks, Fuzzy Systems, Machine learning and probabilistic reasoning etc. Emphasis of this course will be on Artificial Neural Networks, Fuzzy Logic, Meta-heuristic techniques like Genetic Algorithms, ACO and PSO and their applications to different computational problems. Besides the written papers, lab-based examinations are included as part of the assessment requirements for the study. The lab-based examinations will test the candidate's ability to develop computer-programming solutions for a series of computational tasks of varying complexity.

Prerequisite

14CS270 - Problem solving using Computers,

14CS430 - Design and Analysis of algorithms

14CS720 - Artificial Intelligence.

Course Outcomes

On the successful completion of the course, students will be able to

Describe the architecture and learning of various supervised and unsupervised artificial neural networks (CO1)	Understand
Apply neural network learning to pattern classification and regression problems (CO2)	Apply
Explain various parts of fuzzy logic based decision making process	Understand
Describe various processes of Genetic Algorithms such as initialization, selection, reproduction (CO3)	Understand
Describe the basic procedure and parameters involved in Ant Colony Optimization and Particle swarm optimization algorithms (CO4)	Understand
Apply Genetic Algorithms, Ant Colony Optimization and Particle swarm optimization to combinatorial optimization problems (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	S	M						M			
CO2.	S	S	S	S					S			
CO3.	S	S	M						M			
CO4.	S	S	M						M			
CO5.	S	S	M						M			
CO6.	S	S	S	S					S			

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	15	10	10	20
Understand	20	20	20	60
Apply	15	20	20	20
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Enumerate the demerits of back propagation algorithm. (Remember)
2. Differentiate the supervised and unsupervised learning? (Understand)
3. State the training procedure of Kohonen layer and Grossberg layer in counter propagation network. (Understand)

Course Outcome 2 (CO2):

1. Apply neural networks to recognize handwritten characters.
2. Apply neural networks for image compression.
3. Apply neural network for stock market prediction

Course Outcome 3 (CO3):

1. Consider the fuzzy set Young defined by the membership function sig(age; -4, 12). What is the degree of membership value of a student of age 15 in the fuzzy set More or Less Young but not Too Young? (Remember)
2. Consider a fuzzy set A defined by the trapezoidal membership function trapezoid(x; 20, 30, 60, 90). Determine de-fuzzification results using (a) Centroid of Area (b) Bisector of Area. (Understand)
3. Discuss the basic fuzzy sets operation. (Understand)

Course Outcome 4 (CO4):

1. List different selection mechanisms in GA? (Remember)
2. Show how binary encoding can be done for knapsack problem. (Understand)
3. Describe various processes of Genetic Algorithms. (Understand)

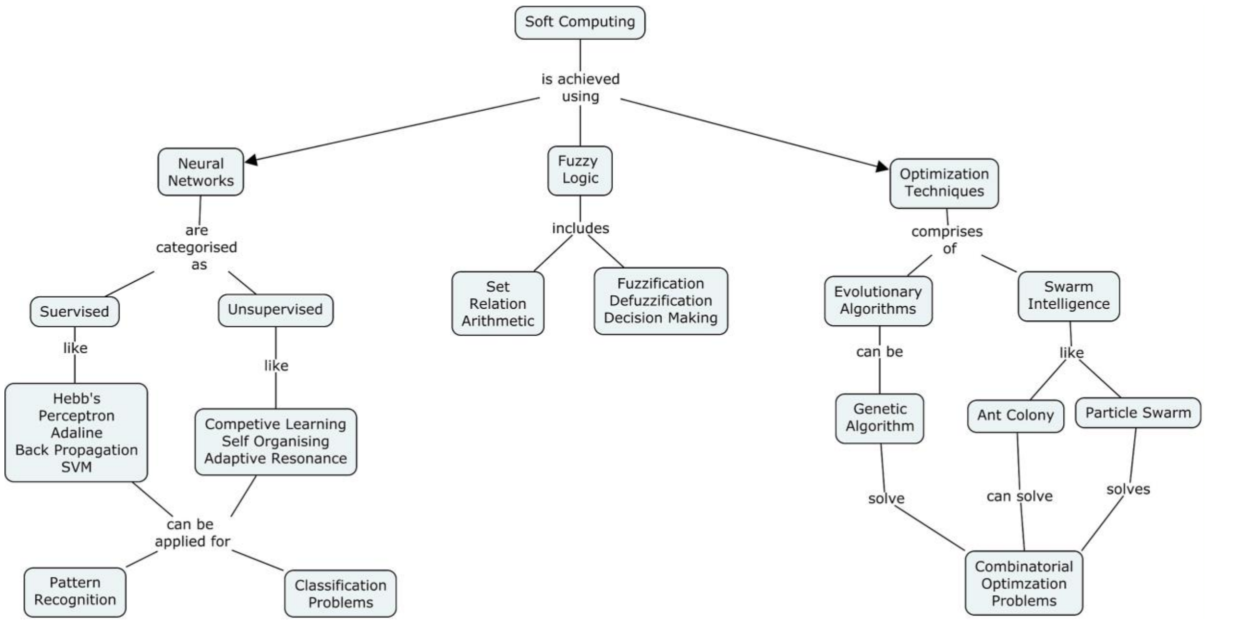
Course Outcome 5 (CO5):

1. Define pbest and gbest in PSO. (Remember)
2. Discuss main steps of the ACO algorithm (Understand)
3. Write the drawbacks of PSO and explain how are they overcome in newer versions?

Course Outcome 6 (CO6):

1. Apply GA for solving MAX -3SAT problems.
2. Apply ACO for solving graph coloring problem
3. Apply PSO for Longest Common Subsequence Problem

Concept Map



Syllabus

Soft Computing: Introduction, requirement, different tools and techniques, usefulness and applications. **Artificial Neural Network**: Supervised Learning Neural Network, Hebb's learning, Perceptron, Adaline, Multilayer feed forward network, Back propagation, Radial Basis function networks, Support Vector Machine, Unsupervised Learning Neural Network, Competitive learning, Self-Organizing Feature Maps, Adaptive Resonance Theory, Case studies on classification and pattern recognition problems-Overview of deep learning. **Introduction to fuzzy logic**, fuzzy sets & relations, fuzzy arithmetic, Fuzzification and defuzzification, fuzzy decision making, Case studies on decision making problems. **Optimization**, introduction to traditional optimization and search techniques, Evolutionary algorithms and search space, General Genetic algorithm, operators, stopping criteria, constraints, Case studies on combinatorial problems. **Swarm Intelligent Algorithms**, Ant Colony Optimization, Cemetery Organization and Brood Care - Particle swarm optimization- Basic PSO, Basic Variations, PSO Parameters, Case studies on combinatorial problems.

Text Books

1. S.N Sivanandam and S.N Deepa, Principles of Soft Computing, Second Ed. Wiley Publishers, 2013.
2. Andries P. Engelbrecht**,** Computational Intelligence: An Introduction, Second Ed, John Wiley & Sons, 2007.
3. Marco Dorigo, Thomas Stu¨ tzle, Ant Colony Optimization, A Bradford Book,The MIT Press Cambridge,2004.
4. James Kennedy and Russell Eberhart, “Particle Swarm Optimization”, Proceedings of IEEE International Conference on Neural Networks, 1995.
5. Russell Eberhart and James Kennedy, “A New Optimizer Using Particle Swarm Theory”, Sixth International Symposium on Micro Machine and Human Science, 1995.
6. Jürgen Schmidhuber, “Deep learning in neural networks: An overview”, Neural Networks, Elsevier, Volume 61, January 2015, Pages 85–117.
7. Case studies - www.ibm.com/analytics/watson-analytics/

Reference Books

1. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989
2. George J. Klir, Bo Yuan Fuzzy Sets and Fuzzy Logic Theory and Applications, Prentice Hall PTR, 1995.
3. Timothy J Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, Wiley Publishers, 2010.

Course Contents and Lecture Schedule for Theory

Module No.	Topic	No. of Lectures
	Soft Computing techniques	
0	Introduction to Soft Computing	1
1	Artificial Neural Network	
1.1	Supervised Learning Neural Network -Hebb's learning, Perceptron,	1
1.1.2	Adaline, Multilayer feed forward network	1
1.1.3	Back propagation & Radial Basis function networks	3
1.1.4	Support Vector Machine	1
1.2	Unsupervised Learning Neural Network- Competitive learning	1
1.2.1	Self-Organizing Feature Maps & Adaptive Resonance Theory	3
1.3	Case studies on classification and pattern recognition problems. Overview of deep learning	2
2	Introduction to fuzzy logic	
2.1	fuzzy sets & relations	1
2.2	fuzzy arithmetic	2
2.3	Fuzzification & defuzzification	2
2.4	fuzzy decision making	2
2.5	Case studies on decision making problems	2
3	Optimization techniques - Evolutionary algorithms	
3.1	introduction to traditional optimization and search techniques, Evolutionary algorithms and search space	2
3.2	General Genetic algorithm, Operators- selection, crossover, mutation	2
3.3	stopping criteria & constraints	1
3.4	Case studies on combinatorial problems	3
4	Optimization techniques - Swarm Intelligent Algorithms	
4.1	Ant Colony Optimization algorithm	1
4.1.1	Cemetery Organization and Brood Care	1
4.2	Particle swarm optimization- Basic PSO, Basic Variations, PSO Parameters	2
4.3	Case studies on combinatorial problems.	2

Course Designer:

1. S. Sudha ssj@tce.edu

14CSPR0	**KERNEL PROGRAMMING**	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

This course introduces basics of designing kernel components in structuring the operating system. The course is structured on a widely used operating system Linux. The students will get a chance to reinforce concepts in the working of a "real" operating system. The idea is to learn and explore a full-fledged operating system and to use it for kernel-based modifications.

Prerequisite

14CS421 : System Software and Operating Systems

Course Outcomes

On the successful completion of the course, students will be able to

Understand the role of kernel in operating system and the types of kernel design (CO1)	Understand
Understand the boot process and initialization of startup scripts in Linux (CO2)	Understand
Configure Linux kernel services using GUI tools and commands (CO3)	Apply
Understand the kernel role in managing processes and file system (CO4)	Understand
Configure the file system and kernel modules of the Linux system (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M		M	M	M							
CO2.					M							
CO3.	S	M	M		M							
CO4.					M							
CO5.	S	M	S		M							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	40	30	20	20
Understand	40	40	40	40
Apply	20	30	40	40
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is system call?
2. List the tasks performed by Linux kernel.
3. How does user mode differ from privileged mode?

Course Outcome 2 (CO2):

1. What is runlevel?
2. Explain the steps in booting of Linux kernel.
3. List the various runlevels provided by Linux.

Course Outcome 3 (CO3):

1. How the printer service is started in Linux?
2. Name the commands used to start or stop Linux services.
3. Describe the GUI tools used for configuring Linux services.

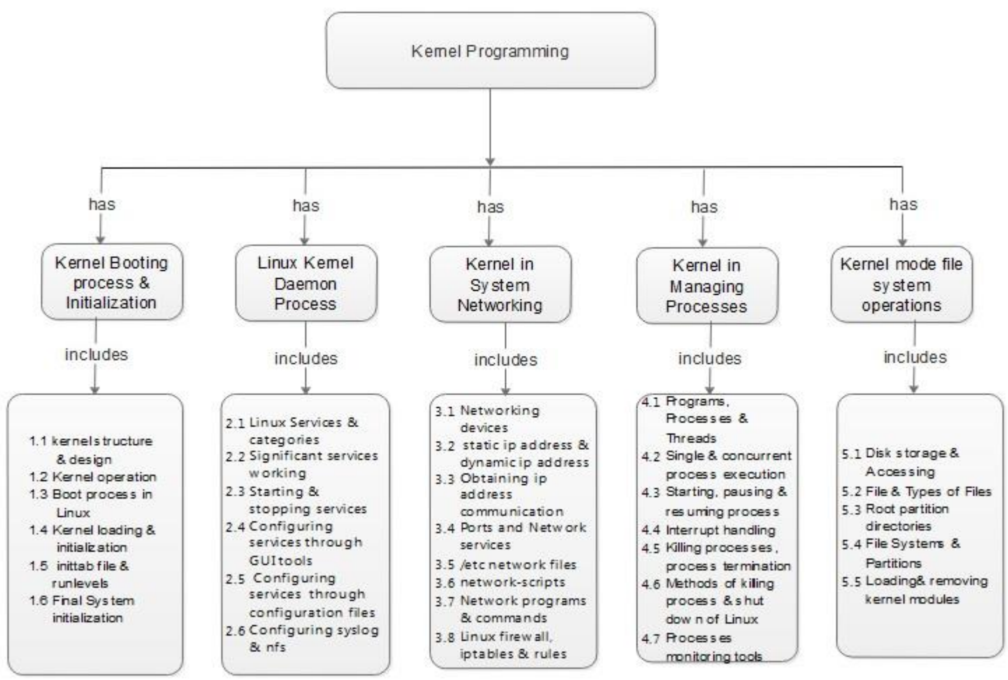
Course Outcome 4 (CO4):

1. How the process execution is monitored in Linux?
2. List the differences between process and thread.
3. How process communicates in Linux?

Course Outcome 5 (CO5):

1. Name the files to work with when configuring the nfs.
2. Explain the file system in Linux.
3. Give the command used to plugin a Linux kernel module.

Concept Map



Syllabus

Kernel Booting process & Initialization:
UNIX kernel structure – Linux kernel design and modes of operation - system calls - the kernel role - boot process - boot loaders in Linux - loading the Linux kernel - initialization of the Linux operating system - inittab file and runlevels - executing rcs.conf and rc.sysinit - rc.conf and rc scripts - finalizing system initialization **Linux Kernel Daemon Process:**
Linux services - categories of services - Significant Linux services working- starting and stopping services - configuring services through GUI tools - configuring services through configuration files - configuring syslog - configuring nfs **Kernel in System Networking:** Networking devices - System ip address - static ip address - dynamic ip address - setting up a DHCP server - the TCP/IP protocol stack - ports - Network services – the /etc/sysconfig/network-scripts directory's contents - other network services - /etc network files - network programs - the ip program - Network commands - the Linux firewall - the iptables - config file - rules for the iptables - examples of firewall rules **Kernel in Managing Processes:** Programs, Processes and Threads - Process address space - single process execution - concurrent processing - starting, pausing, and resuming processes - ownership of running processes - launching processes from a shell - interrupt handling - managing Linux processes - killing processes, process termination, methods of killing processes and methods to shut down Linux - monitoring processes - GUI monitoring tools, command-line monitoring tools **Kernel mode file system operations:** Disk storage and Accessing – File and Types of Files - Root partition directories - /dev, /proc, /sys, /etc, /home, /usr and /var – File Systems – mount points, mounting and unmounting file systems – Partitions – Device drivers - loading and removing kernel modules

Text Books

1. Linux with Operating System Concepts, Richard Fox, CRC Press, Taylor & Francis, A Chapman & Hall Book.
2. Linux Kernel Development, Robert Love, Pearson Education, Third Edition.

Reference Books

1. Advanced Programming in the UNIX environment, W.Richard Stevens, Stephen A.Rago, Addison-Wesley Professional Computing Series, Third Edition.
2. The Linux Kernel Module Programming Guide, Peter Jay Salzman, Michael Burian, Ori Pomerantz
3. Linux System Programming, Robert Love, O'Reilly Media, Inc., Second Edition.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Kernel Booting process & Initialization**	
1.1	UNIX kernel structure and Linux kernel design	1
1.2	Kernel tasks, modes of operation and system calls	1
1.3	Boot process and boot loaders in Linux	1
1.4	Kernel loading and initialization	1
1.5	inittab file and runlevels	2
1.6	rcs.conf and rc.sysinit,scripts and final system initialization task	1
2.	**Linux Kernel Daemon Process**	
2.1	Linux Services and categories of services	1
2.2	Significant Linux services working	1

Module No.	Topic	No. of Lectures
2.3	Starting and stopping services	1
2.4	Configuring services through GUI tools	1
2.5	Configuring services through configuration files	1
2.6	Configuring syslog	1
2.7	Configuring nfs	1
3.	**Kernel in System Networking**	
3.1	Networking devices	1
3.2	System ip address, static ip address and dynamic ip address	1
3.3	Obtaining ip address by setting up a DHCP server and the TCP/IP protocol stack communication	1
3.4	Ports and Network services	1
3.5	/etc network files	1
3.6	/etc/sysconfig/network-scripts directory's contents	1
3.7	Network programs, the ip program and network commands	1
3.8	Linux firewall, the iptables, rules for the iptables and examples of firewall rules	2
4.	**Kernel in Managing Processes**	
4.1	Programs, Processes and Threads and Process address space	1
4.2	Single process execution and concurrent processing	1
4.3	Starting, pausing, and resuming processes - ownership of running processes - launching processes from a shell	1
4.4	Interrupt handling	1
4.5	Killing processes, process termination,	1
4.6	Methods of killing processes and methods to shut down Linux	1
4.7	Monitoring processes - GUI monitoring tools, command-line monitoring tools	1
5.	**Kernel mode file system operations**	
5.1	Disk storage and Accessing	1
5.2	File and Types of Files	1
5.3	Root partition directories :/dev, /proc, /sys, /etc, /home, /usr and /var	1
5.4	File Systems – mount points, mounting and unmounting file systems, partitions and device drivers	2
5.5	Loading and removing kernel modules	1
	Total No. of hours	36

Course Designer:

1. G.Madhu Priya gmadhupriya@tce.edu

		Category	L	T	P	Credit
14CSPS0	**NETWORK DIRECTORY SERVICES**	PE	3	0	0	3

Preamble

This course aims at exploring the features of directory services and the logical and physical administration of active directory. It is used to centrally store and manage security principals, such as users, groups, and computers, and it offers centralized and secure access to network resources. An overview of backup and restore services of active directory is also provided.

Course Outcomes

On the successful completion of the course, students will be able to

Demonstrate a deep understanding of the common networking services like DNS and DHCP (CO1)	Apply
Describe and configure server roles with Active Directory Services in Windows Server 2008 (CO2)	Understand
Deploy Active Directory Domain Services (CO3)	Apply
Explore the physical components of Active Directory (CO4)	Understand
Manage accounts, subnets, Site-Links, Group Policy, and DNS configuration with Active Directory Domain Services (CO5)	Apply
Examine backup and restore features in Active Directory Domain Services(CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	M			M							
CO2.	S	S										
CO3.	S	S										
CO4.	S	S										
CO5.	S	S			M							
CO6.	S	S	S		M							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
Remember	30	20	20	20
Understand	30	40	40	30
Apply	40	40	40	50

Analyze	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. State the need for DNS and DHCP. (Remember)
2. Draw the DHCP architecture (Understand)
3. Explain DHCP lease process. (Understand)

Course Outcome 1 (CO2):

1. Define : Microsoft Active Directory?.(Remember)
2. Define: LDAP.(Remember)
3. Explain the configuration process of ADDS Services. (Understand)

Course Outcome 2 (CO3):

1. State the need for Group policy**.(**Remember)
2. Explain the different services in ADDS. (Understand)
3. Examine the steps involved in installation of ADDS.(Apply)

Course Outcome 3 (CO4):

1. State the need for ADDS. (Remember)
2. List the components of ADDS. (Remember)
3. Explain the salient features of ADDS. (Understand)

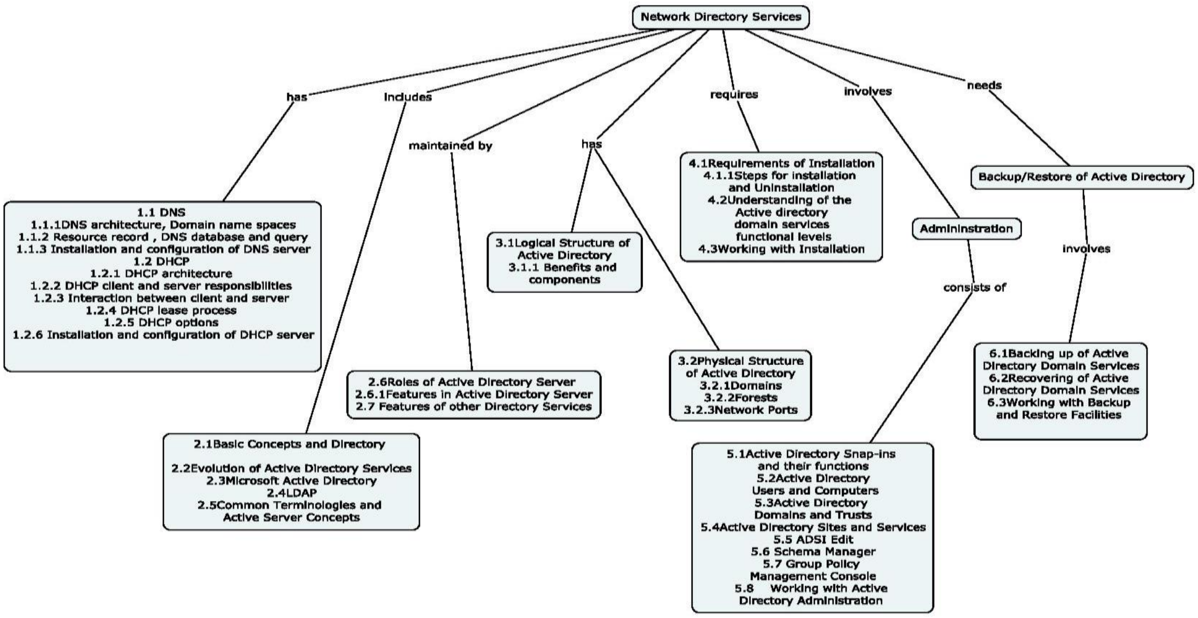
Course Outcome 4 (CO5):

1. State the need for subnet.(Remember)
2. Explain the need for DNS. (Understand)
3. Examine the process to create users and organizational units in active directory? (Apply)

Course Outcome 5 (CO6):

1. State the need for connectors.(Remember)
2. Explain the process for performing replication. (Understand)
3. Examine the process of backup of active directory in windows server 2008? Illustrate with an example. (Apply)

Concept Map



Syllabus

Information and Resource sharing: DNS, DNS architecture, Domain name spaces, Resource record , DNS database and query, Installation and configuration of DNS server, **DHCP**, DHCP architecture ,DHCP client and server responsibilities, Interaction between client and server, DHCP lease process, DHCP options, Installation and configuration of DHCP server
Introduction to Directory - Basic Concepts, Directory, Evolution of Active Directory Services, Microsoft Active Directory, LDAP, Common Terminologies and Active Server Concepts, Active Directory Structure and Storage Technologies, Roles of Active Directory Server, Features in Active Directory Server, Features of other Active Directory Services. **Active Directory Domains and Forests** - Logical Structure of Active Directory, Benefits and components, Physical Structure of Active Directory, Domains, Forests, Network Ports. **Installation of Active Directory Domain Services in Windows 2008 R2** – Requirements for Installation, Steps for installation and Uninstallation, Understanding of the Active directory domain services functional levels, working with Installation. **Active Directory Administration** - Active Directory Snap-ins and their functions, Active Directory Users and Computers, Active Directory Domains and Trusts, Active Directory Sites and Services, ADSI Edit, Schema Manager, Group Policy Management Console, Working with Active Directory Administration. **Backup/ Restore of Active Directory** - Backing up of Active Directory Domain Services, Recovering of Active Directory Domain Services, Working with Backup and Restore Facilities.

References

1. http://technet.microsoft.com/en-us/library/cc787921(v=ws.10).aspx – What is DNS?
2. http://technet.microsoft.com/en-us/library/cc772774(v=ws.10).aspx – How DNS Works
3. http://technet.microsoft.com/en-us/library/cc775464(v=ws.10).aspx – DNS Tools and Settings

4. http://technet.microsoft.com/en-us/library/cc725925.aspx - Install a DNS Server in Windows Server 2008 R2
5. http://technet.microsoft.com/en-us/library/cc731053(WS.10).aspx – Active Directory Domain Services Overview
6. http://technet.microsoft.com/en-us/library/cc773108(v=WS.10).aspx – Operations Master Roles
7. http://technet.microsoft.com/en-us/library/cc759186(v=WS.10).aspx - Active Directory Structure and Storage Technologies
8. http://technet.microsoft.com/en-us/library/cc754697.aspx - Understanding Sites, Subnets and Site Links
9. http://technet.microsoft.com/en-us/library/cc759073(v=ws.10).aspx
10. http://technet.microsoft.com/en-us/library/cc783351(v=ws.10).aspx#w2k3tr_logic_how_rqma
11. http://technet.microsoft.com/en-us/library/cc771433(v=ws.10).aspx - Scenarios for Installing AD DS
12. http://technet.microsoft.com/en-us/library/cc771188(v=ws.10).aspx - Requirements for installing AD DS
13. http://technet.microsoft.com/en-us/library/cc772464(v=ws.10).aspx - Insalling a New Forest
14. http://technet.microsoft.com/en-us/library/understanding-active-directory-functional-levels(v=ws.10).aspx - Understanding Active Directory Domain Services Functional Levels
15. http://technet.microsoft.com/en-us/library/cc753298.aspx - Group Policy Management Console
16. http://technet.microsoft.com/en-us/library/cc816584(v=WS.10).aspx - Backing Up Active Directory Domain Services
17. http://technet.microsoft.com/en-us/library/cc816751(v=WS.10).aspx - Recovering Active Directory Domain Services
18. http://technet.microsoft.com/en-us/library/cc794908(v=ws.10).aspx - Administering Active Directory Domain Services
19. http://technet.microsoft.com/en-us/library/cc754217.aspx - Active Directory Users and Computers
20. http://technet.microsoft.com/en-us/library/cc770299.aspx - Active Directory Domains and Trusts
21. http://technet.microsoft.com/en-us/library/cc730868.aspx - Active Directory Sites and Service
22. http://technet.microsoft.com/en-us/library/cc730667.aspx - Schema Manager

Text Books

1. Timothy A. Howes Ph.D. (Author), Mark C. Smith , Understanding and Deploying LDAP Directory Services Hardcover – 2nd edition ISBN-13: 978-0672334467.

2. Brian Desmond,Joe Richards, Robbie Allen, Alistair G.. Active Directory, Designing, Deploying, and Running Active Directory - Lowe-Norris Publisher: O'Reilly Media 5th Edition.

3. Jeremy Moskowitz, Group Policy: Fundamentals, Security, and the Managed Desktop 3rd Edition.

Course Contents and Lecture Schedule

1.	**Information and Resource sharing**	
1.1	**DNS**	
1.1.1	DNS architecture, Domain name spaces	1
1.1.2	Resource record , DNS database and query	1
1.1.3	Installation and configuration of DNS server	2
1.2	**DHCP**	
1.2.1	DHCP architecture	1
1.2.2	DHCP client and server responsibilities	
1.2.3	Interaction between client and server	
1.2.4	DHCP lease process	1
1.2.5	DHCP options	
1.2.6	Installation and configuration of DHCP server	2
2	**Introduction to Microsoft Active Directory**	
2.1	Basic Concepts and Directory	1
2.2	Evolution of Active Directory Services	1
2.3	Microsoft Active Directory	1
2.4	LDAP	1
2.5	Common Terminologies and Active Server Concepts	1
2.6	Roles of Active Directory Server	2
2.6.1	Features in Active Directory Server	
2.7	Features of other Directory Services	
3	**Active Directory Domains and Forests**	
3.1	Logical Structure of Active Directory	2
3.1.1	Benefits and components	
3.2	Physical Structure of Active Directory	3
3.2.1	Domains	
3.2.2	Forests	
3.2.3	Network Ports	1
4	**Installation of Active Directory Domain Services in Windows 2008 R2**	
4.1	Requirements of Installation	2
4.1.1	Steps for installation and Uninstallation	
4.2	Understanding of the Active directory domain services functional levels	2
4.3	Working with Installation	2
5	**Active Directory Administration**	

5.1	Active Directory Snap-ins and their functions	1
5.2	Active Directory Users and Computers	2
5.3	Active Directory Domains and Trusts	
5.4	Active Directory Sites and Services	
5.5	ADSI Edit	2
5.6	Schema Manager	
5.7	Group Policy Management Console	
5.8	Working with Active Directory Administration	1
6	**Backup/Restore of Active Directory**	
6.1	Backing up of Active Directory Domain Services	1
6.2	Recovering of Active Directory Domain Services	1
6.3	Working with Backup and Restore Facilities	1
	TOTAL	**36**

Course Designers:

1. Mr. G. Shashi Kumar, Escalation Engineer, Microsoft GTSC, Bangalore, shashikg@microsoft.com

2. C.Deisy, cdcse@tce.edu

3. C.Senthilkumar cskcse@tce.edu

		Category	L	T	P	Credit
14CSPT0	**APPLIED MACHINE LEARNING**	PE	3	0	0	3

Preamble

Machine Learning is the broad discipline of teaching computers to perform tasks without explicitly programming them. Now-a-days machine learning is employed to perform a variety of tasks which a human alone could do a couple of years ago like driving cars, object recognition, speech recognition, playing games, optical character recognition etc. In this course students will learn a spectrum of machine learning algorithms with a sound math-based theoretical background along with the technical know-how of applying these algorithms to perform a variety of tasks to build applications.

Prerequisites

14MA110 - Engineering Mathematics - I (Linear Algebra and Calculus)

14CS210 - Engineering Mathematics-II (Calculus)

14CS270 - Problem Solving using Computers

14CS310 - Probability and Statistics

14CS410 - Discrete Mathematics and Combinatorics

Course Outcomes

On the successful completion of the course, students will be able to

Construct algorithms to learn linear regression models to predict the value of a continuous-valued output given a training data consisting of uni-variate/multi-variate input features. (CO1)	Apply
Develop learning algorithms based on logistic regression, Support Vector Machines to predict discrete-valued output given a training data comprising of features and corresponding class labels. (CO2)	Apply
Construct algorithms based on neural networks to perform simple learning tasks like speech recognition, digit recognition, optical character recognition and similar cognitive applications. (CO3)	Apply
Develop unsupervised learning algorithms to learn patterns from given training set of unlabeled data points. (CO4)	Apply
Given a model and an application for which the model has been built, assess the performance of the model using model validity indices. (CO5)	Evaluate
Construct implementations to program solutions for real-world machine learning problems like digit recognition, speech recognition, object recognition and optical character recognition using the toolbox provided by R/Octave and other Artificial Intelligence frameworks. (CO6)	Apply
Construct reinforcement learning algorithms for modelling trial and error based adaptive learning for environments where an explicit control instruction cannot be provided. (CO7)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	S	S	M								
CO2	S	S	S	M								
CO3	S	S	S	M								
CO4	S	S	S	M								
CO5	S	S	S	M								
CO6	S	S	S	S	S							M
CO7	S	S	S	M								

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	0	0	0	0
Understand	15	10	10	20
Apply	35	40	30	60
Analyse	0	0	0	0
Evaluate	0	0	10	20
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Consider a data set in which each data point t_n is associated with a weighting factor $r_n > 0$, so that the sum-of-squares error function becomes:

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N} r_n \left(t_n - w^T\phi(x_n)\right)^2$$

Construct an expression for the solution w* that minimizes this error function. Give two alternative interpretations of the weighted sum-of-squares error function in terms of (i) data dependent noise variance and (ii) replicated data points.

2. Construct a linear regression model to fit the given house-price prediction training data. The task is to predict the price of a house given its features as input. Your model should generalize well to the test dataset.

3. We have seen that, as the size of a data set increases, the uncertainty associated with the posterior distribution over model parameters decreases. Make use of the matrix identity:

$$\left(M + vv^T\right)^{-1} = M^{-1} - \frac{(M^{-1}v)\left(v^T M^{-1}\right)}{1 + v^T M^{-1} v}$$

to show that the uncertainty $\sigma_N^2(x)$ associated with the linear regression function satisfies $\sigma_N^2(x) \leq \sigma_{N+1}^2(x)$

Course Outcome 2 (CO2):

1. Show that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector **w** whose decision boundary $\mathbf{w}^T \phi(\mathbf{x}) = 0$ separates the classes and then taking the magnitude of **w** to infinity.

2. Using the expression for the derivative of the logistic sigmoid, construct an expression for the derivative of the error function for the logistic regression model.

3. Consider a binary classification problem in which each observation $\mathbf{x}_n$ is known to belong to one of two classes, corresponding to $t = 0$ and $t = 1$, and suppose that the procedure for collecting training data is imperfect, so that training points are sometimes mislabelled. For every data point $\mathbf{x}_n$, instead of having a value t for the class label, we have instead a value π_n representing the probability that $t_n = 1$. Given a probabilistic model $p(t = 1|\phi)$, construct an expression for the log likelihood function appropriate to such a data set.

Course Outcome 3 (CO3)

1. Consider a binary classification problem in which the target values are $t \in \{0, 1\}$, with a network output $y(\mathbf{x},\mathbf{w})$ that represents $p(t = 1|\mathbf{x})$, and suppose that there is a probability ϵ that the class label on a training data point has been incorrectly set. Assuming independent and identically distributed data, write down the error function corresponding to the negative log likelihood. Verify that the error function is obtained when $\epsilon = 0$. Note that this error function makes the model robust to incorrectly labelled data, in contrast to the usual error function.

2. Show that maximizing likelihood for a multiclass neural network model in which the network outputs have the interpretation $y_k(\mathbf{x},\mathbf{w}) = p(t_k = 1|\mathbf{x})$ is equivalent to the minimization of the cross-entropy error function.

3. Consider a neural network, such as the convolutional network discussed in class, in which multiple weights are constrained to have the same value. Discuss how the standard backpropagation algorithm must be modified in order to ensure that such constraints are satisfied when evaluating the derivatives of an error function with respect to the adjustable parameters in the network.

Course Outcome 4 (CO4)

1. Consider the K-means algorithm discussed in class. Show that as a consequence of there being a finite number of possible assignments for the set of discrete indicator variables *rnk*, and that for each such assignment there is a unique optimum for the $\{\mu_k\}$, the K-means algorithm must converge after a finite number of iterations.

2. Consider a special case of a Gaussian mixture model in which the covariance matrices $\mathbf{\Sigma}_k$ of the components are all constrained to have a common value $\mathbf{\Sigma}$. Derive the EM equations for maximizing the likelihood function under such a model.

3. Verify that maximization of the complete-data log likelihood for a Gaussian mixture model leads to the result that the means and covariances of each component are fitted independently to the corresponding group of data points, and the mixing coefficients are given by the fractions of points in each group.

Course Outcome 5 (CO5)

1. Consider two classifiers to solve a handwritten digit recognition problem. Compare the predictive accuracy of the two classifiers and determine the better of the two using statistical methods.

2. Consider a classification problem in which the loss incurred when an input vector from class C_k is classified as belonging to class C_j is given by the loss matrix L_{kj}, and for which the loss incurred in selecting the reject option is λ. Find the decision criterion that will give the minimum expected loss. Verify that this reduces to the reject criterion discussed in class when the loss matrix is given y $L_{kj} = 1 - I_{kj}$. Determine the relationship between λ and the rejection threshold θ?

3. Assess how confidence intervals are helpful in gauging the goodness of the test error approximation.

Course Outcome 6 (CO6)

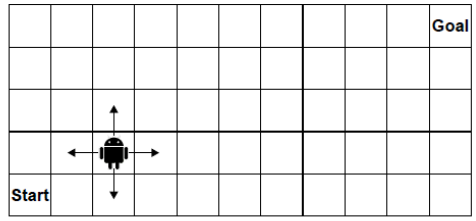
1. Construct and implement a neural network in Octave to predict the value of an input handwritten digit. The training set will be provided as a set of images containing handwritten digits.

2. Construct and implement a logistic regression based mail classification system to categorize an incoming email into one of the following: Personal, Work, Movies, Social Circles, Promotions and Spam.

3. Construct and implement an unsupervised learning algorithm based on expectation-maximization to segment an input image into its constituent objects.

Course Outcome 7 (CO7)

1. Consider the following scenario: An agent is situated in a 11x5 grid environment as illustrated in the figure below and has to learn the shortest path from all possible positions to the goal area. For your experiments please use the following reward scheme: After each action, the agent is rewarded 0 if the action ends in the target area and -1 otherwise. At the beginning of each episode, the agent is placed on a randomly chosen position. Possible actions are moving one cell in one of the four cardinal directions. If an agent tries to leave the world, it remains on its position. An episode ends either if the agent has reached the goal region or after i_{max} iterations. Use an appropriate exploration strategy.



Solve the following:

1. What is the impact of the given reward function? How does it influence the strategy (policy) of the agent for finding the goal?

2. Describe the elements of the underlying MDP
(i) What is the state set?
(ii) What is the action set?
(iii) Give a formal definition of the described reward function.
(iv) How does the state transition function look like (informally)?

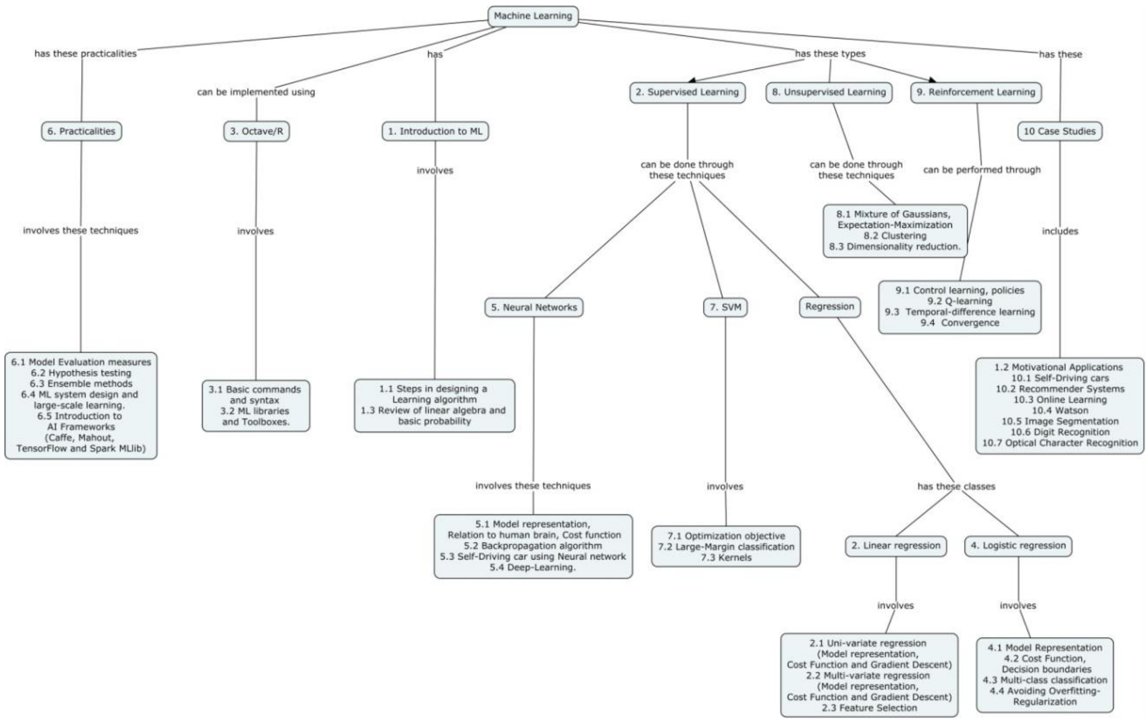
3. Is it sufficient to let the agent learn for a single episode only, or should the agent learn for several episodes in a row? Give reasons for your answer.

4. Implement the single agent reinforcement learning problem using deterministic Q-Learning as described in Mitchell's book

5. Consider the following two approaches for measuring the performance of the agent:
a) The accumulated reward at the end of each learning episode.
b) Position the agent randomly in the environment and follow the greedy strategy until either the goal is reached or the iteration limit of i_{max} iterations is exceeded. Repeat this procedure n times and use the average accumulated reward as performance measure.

What are the advantages/disadvantages of these two approaches? Select and implement one of them and explain your choice.

Concept Map



Syllabus

Introduction to Machine Learning: Steps in designing a learning algorithm, Motivational Applications (Watson, Self-driving cars, Recommender systems), Review of linear algebra and basic probability.
Linear Regression: Uni-variate regression, Multi-variate regression, Model representation, Cost Function, Gradient Descent, Feature Selection.
Octave/R: Basic commands and syntax, ML libraries and Toolboxes.
Logistic Regression: Model Representation, Cost Function, Decision boundaries, Multi-class classification. Avoiding Overfitting-Regularization.
Neural Networks: Model representation, Relation to human brain, Cost function, Backpropagation algorithm, Self-Driving car using Neural network, Deep-Learning.
Practicalities: Model Evaluation measures, Hypothesis testing, Ensemble methods, ML system design and large-scale learning, Introduction to AI Frameworks (Caffe, Mahout, TensorFlow and Spark MLlib)
Support Vector Machines: Optimization objective, Large-Margin classification, Kernels.
Unsupervised Learning: Mixture of Gaussians, Expectation-Maximization, Clustering, Dimensionality reduction.
Reinforcement Learning: Control learning, policies, Q-learning, Temporal-difference learning, Convergence.
Case Studies: Self-Driving cars, Recommender Systems, Online Learning, Watson, Image Segmentation, Digit Recognition, Optical Character Recognition.

Text Books

1. Ethem Alpaydin, “Introduction to Machine Learning”, PHI, Third edition, 2015.
2. Christopher M.Bishop, “Pattern recognition and machine learning”, Springer, 2007.

Reference Books

1. Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie, “The Elements of Statistical Learning”, Springer, 2011. (Available for download on the author’s web-page: http://statweb.stanford.edu/~tibs/ElemStatLearn/)

2. Tom M. Mitchell, “Machine learning”, McGraw Hill, 1997.
3. Kevin Murphy, “Machine Learning - A Probabilistic Perspective, Adaptive Computation and Machine Learning”, MIT Press, 2012.
4. Stephen Marsland, “Machine learning: An algorithmic perspective”, CRC, 2009.
5. R. S. Sutton and A. G. Barto, “Reinforcement Learning - An Introduction”, MIT Press, 1998.
6. Stuart Russell, Peter Norvig, “Artificial Intelligence: A Modern Approach”, Prentice Hall, third edition, 2009.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	Introduction to Machine Learning	
1.1	Steps in designing a learning algorithm	1
1.2	Motivational Applications (Watson, Self-driving cars, Recommender systems)	1
1.3	Review of linear algebra and basic probability.	2
2.	Linear Regression	
2.1	Uni-variate regression (Model representation, Cost Function and Gradient Descent)	1
2.2	Multi-variate regression (Model representation, Cost Function and Gradient Descent)	1
2.3	Feature Selection	1
3	Octave/R	
3.1	Basic commands and syntax	1
3.2	ML libraries and Toolboxes.	1
4	Logistic Regression	
4.1	Model Representation	1
4.2	Cost Function, Decision boundaries	1
4.3	Multi-class classification	1
4.4	Avoiding Overfitting-Regularization	1
5	Neural Networks	
5.1	Model representation, Relation to human brain, Cost function	1
5.2	Backpropagation algorithm	1
5.3	Self-Driving car using Neural network	1
5.4	Deep-Learning.	1
6	Practicalities	
6.1	Model Evaluation measures	1
6.2	Hypothesis testing	1
6.3	Ensemble methods	1
6.4	ML system design and large-scale learning.	1
6.5	Introduction to AI Frameworks (Caffe, Mahout, TensorFlow and Spark MLlib)	1
7	Support Vector Machines	
7.1	Optimization objective	1
7.2	Large-Margin classification	1
7.3	Kernels	1
8	Unsupervised Learning	
8.1	Mixture of Gaussians, Expectation-Maximization	1

Module No.	Topic	No. of Lectures
8.2	Clustering	1
8.3	Dimensionality reduction.	1
9	Reinforcement Learning	
9.1	Control learning, policies	1
9.2	Q-learning	1
9.3	Temporal-difference learning	1
9.4	Convergence	
10	Case Studies	
10.1	Self-Driving cars	1
10.2	Recommender Systems	1
10.3	Online Learning	
10.4	Watson	1
10.5	Image Segmentation	1
10.6	Digit Recognition	
10.7	Optical Character Recognition	1
	Total	**36**

Course Designer:

1. Mr. Karthick Seshadri skcse@tce.edu

14CSPU0	VIRTUAL REALITY	Category	L	T	P	Credit
		PE	3	0	0	3

Preamble

The course will cover principles, architectures, modeling of virtual reality and also providing the knowledge of virtual reality programming for solving the real time problems.

Prerequisite

14CS330 - Computer Graphics

Course Outcomes

On the successful completion of the course, students will be able to

Explain the need, components, benefits and disadvantages of Virtual Reality. (CO1)	Understand
Demonstrate the working of the input and output devices that are interfaced with virtual reality Environments.(CO2)	Understand
Explain the need of rendering and working of various computing architectures for virtual reality. (CO3)	Understand
Design and implement the suitable virtual reality modeling techniques for the given problem. (CO4)	Apply
Develop virtual reality programming concepts with toolkits for the given application. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO7.	L	L										
CO8.	M	L		L								
CO3	M	L	M	L								
CO4	S	S	M	M	M			M	L	L		M
CO5	S	S	M	M	M		M	M	L	L		M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	50	50	30	30
Understand	40	30	40	40
Apply	10	20	30	30
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. State the need of Virtual Reality (VR).
2. List out the benefits of VR.
3. What direct effects can VE immersion have on users?
4. Discuss the direct Effects of VR Simulations on Users.

Course Outcome 2 (CO2):

1. Difference between an absolute and a relative position input device?
2. How do AC magnetic trackers function? How is their accuracy affected by the presence of metal objects and distance from the source?
3. Discuss the advantages and disadvantages of ultrasonic trackers?
4. Explain the working of 5DT Data Glove.
5. How does touch feedback differ from force feedback?
6. Describe the functioning of 3D audio chips. Make a drawing and explain.

Course Outcome 3 (CO3):

1. List out the three stages of an OpenGL rendering pipe and how can their load be balanced?
2. Describe the working of XBox architecture.
3. Consider two computers that render 500,000 polygons/sec and 1 million polygons/sec, respectively. What is their frame rate for a scene containing 25,000 polygons? What is the maximum scene complexity allowed by each machine if the refresh rate needs to be 30 frames/sec?
4. In a single-user distributed architecture, how is the graphics refresh rate decoupled from the haptics one, and why?
5. Draw the network topologies used in multiuser VR for unicast LAN. Explain how updates are transmitted from user to user in each case.

Course Outcome 4 (CO4):

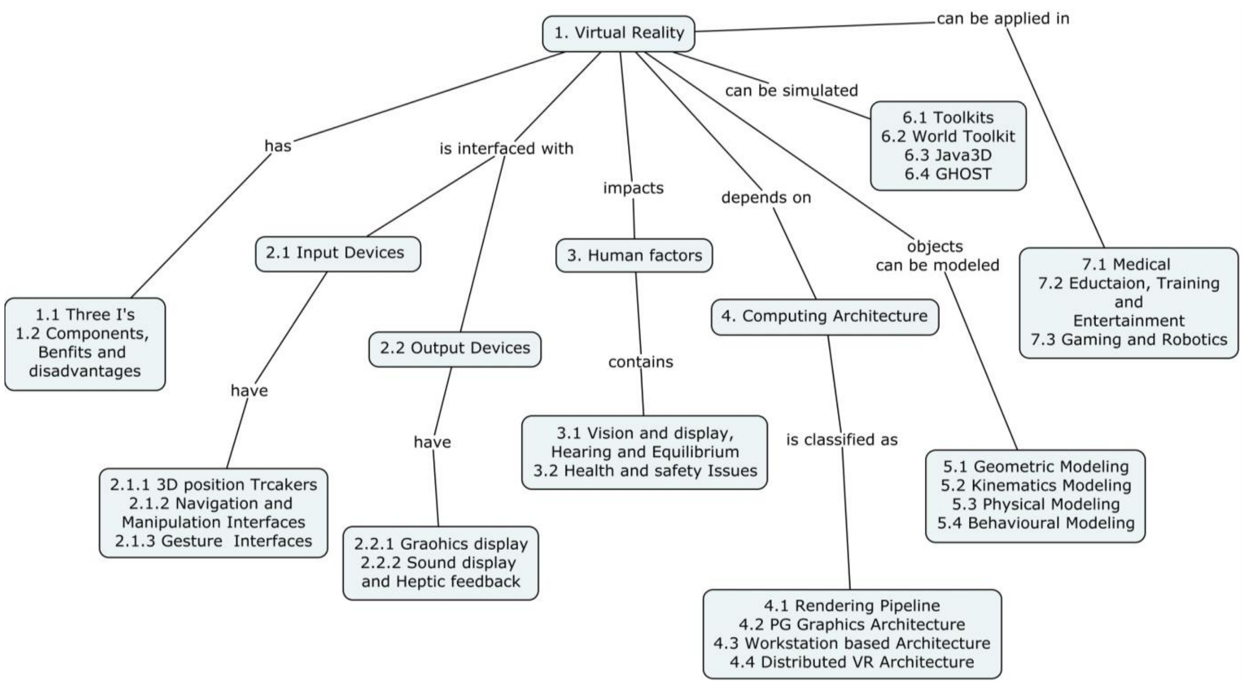
1. What is the difference between local and global illumination? How can global illumination be made interactive?
2. Give examples of multi texturing and explain their importance.
3. How do the projection, clipping, and screen mapping sub stages of the geometry pipeline stage work? Make a drawing and explain.
4. How do you model contact forces for an elastic object that is homogeneous? How about one that has a harder interior kernel?

Course Outcome 5 (CO5):

1. How can a virtual camera be moved in a WTK scene? Make a drawing and explain.
2. Suppose two users observe the same scene on networked computers. What communication mechanism is used by WTK to transmit changes in the camera viewpoint over the net? How can one user have exclusive (but temporary) control over the camera?

3. Draw the GHOST scene graph for a robot and write a program for activation.
4. Why is ADHD assessment in a virtual classroom more realistic than current questionnaire based methods?.
5. List out the applications of VR.
6. Describe the use of VR in military applications with suitable diagram. Assume data wherever necessary.
7. What is a virtual cockpit? Give examples.

Concept Map



Syllabus

INTRODUCTION TO VIRTUAL REALITY: Introduction to Virtual Reality – Three I's of Virtual Reality – Virtual Reality Vs 3D Computer Graphics –- Components of Virtual Reality System - Benefits of Virtual Reality - Disadvantages of Virtual Reality. **VR Hardware :** Input Devices – 3D Position Trackers -Performance Parameters – Types of Trackers - Navigation and Manipulation Interfaces – Gesture Interfaces – Types of Gesture Input Devices. Output Devices – Graphics Display – Human Visual System – Personal Graphics Displays – Large Volume Displays – Sound Displays – Haptic Feedback. **Human Factors :** Vision and Display -Hearing, Tactile and Equilibrium - Health and Safety Issues.**VR ARCHITECTURE :** Computing Architectures of VR – Rendering Principle – Graphics and Haptics Rendering –PC Graphics Architecture – Graphics Accelerators – Graphics Benchmarks – Workstation Based Architectures – Distributed VR Architectures – Multi pipeline Synchronization – Distributed Virtual Environments. **VR MODELING:** Modeling – Geometric Modeling – Virtual Object Shape – Object Visual Appearance – Kinematics Modeling – Transformation Matrices – Object Position – Transformation Invariants –Object Hierarchies – Viewing the 3D World – Physical Modeling – Collision Detection – Surface Deformation – Force Computation – Force Smoothing and Mapping – Behaviour Modeling. **VR PROGRAMMING:** VR Programming – Toolkits and Scene

Graphs – World Toolkit – Java 3D – Comparison of World Toolkit and Java 3D - General Haptics Open Software Toolkit (GHOST**). VR APPLICATIONS:** Medical Applications of VR – Education, Training and Entertainment – Applications of VR in Gaming and Robotics.

Text Books

1. Matjaz Mihelj , Domen Novak, Samo Begus, "Virtual Reality Technology and Applications", 1st Edition, Springer Netherlands, 2014.
2. Grigore C. Burdea, Philip Coiffet, "Virtual Reality Technology", 2nd Edition, Wiley India, 2006.

Reference Books

1. William R.Sherman, Alan B.Craig, "Understanding Virtual Reality – Interface, Application, Design", The Morgan Kaufmann Series, 2003.
2. John Vince, "Introduction in Virtual Reality", Springer, 2004.
3. Gerard Jounghyun Kim, "Designing Virtual Reality Systems, the Structured Approach"，Springer London, 2005.

Course Contents and Lecture Schedule

Module No.	Topic	No. Of Lectures
1.	**Introduction to Virtual Reality**	
1.1	Three I's of Virtual Reality, Virtual Reality Vs 3D Computer Graphics	1
1.2	Components, Benefits and disadvantages of Virtual Reality	1
2.	**VR Hardware**	
2.1	Input Devices	1
2.1.1	3D Position Trackers : Performance Parameters – Types of Trackers	1
2.1.2	Navigation and Manipulation Interfaces	1
2.1.3	Gesture Interfaces – Types of Gesture Input Devices	1
2.2	Output Devices	1
2.2.1	Graphics Display : Human Visual System, Personal Graphics Displays and Large Volume Displays	1
2.2.2	Sound Displays and Haptic Feedback	1
3.	**Human Factors**	
3.1	Vision and Display, Hearing, Tactile and Equilibrium	1
3.2	Health and Safety Issues	1
4.	**VR Architecture**	
4.1	Rendering Principle: Graphics and Haptics Rendering	1
4.2	PC Graphics Architecture: Graphics Accelerators – Graphics Benchmarks	2
4.3	Workstation Based Architectures	2
4.4	Distributed VR Architectures : Multi pipeline Synchronization and Distributed Virtual Environments	2

Module No.	Topic	No. Of Lectures
5.	**VR Modeling**	
5.1	Geometric Modeling : Virtual Object Shape, Object Visual Appearance	1
5.2	Kinematics Modeling	1
5.2.1	Transformation Matrices , Object Position and Invariants	1
5.2.2	Object Hierarchies and Viewing the 3D World	1
5.3	Physical Modeling :Collision Detection, Surface Deformation	1
5.3.1	Force Computation , Force Smoothing and Mapping	1
5.4	Behaviour Modeling	1
6	**VR Programming**	
6.1	Toolkits and Scene Graphs	2
6.2	World Toolkit	2
6.3	Java 3D ,Comparison of World Toolkit and Java 3D	2
6.4	General Haptics Open Software Toolkit (GHOST)	2
7.	**VR Applications**	
7.1	Medical Applications of VR	1
7.2	Education, Training and Entertainment	1
7.3	Applications of VR in Gaming and Robotics	1
	Total	**36**

Course Designer:

1. Dr. S.Sridevi sridevi@tce.edu

		Category	L	T	P	Credit
14CSPV0	**DESIGN PATTERNS**	PE	3	0	0	3

Preamble

This course is indented to cover various software design patterns. The course covers the rationale and benefits of object-oriented software design patterns. Numerous problems will be studied to investigate the implementation of good design patterns

Prerequisite

Nil

Course Outcomes

CO1. Explain what specific object oriented design problem the pattern solves. (Understand)

CO2. Perform the analysis of the software-to-be-developed using an object oriented approach. (Apply)

CO3. Prepare the refined list of entities, their attributes and relationships, design the object types and their interfaces, concrete classes and types for the software-to-be developed (Apply) (Apply)

CO4. Implement the pattern in Java to a real world problem. (Apply)

CO5. Identify the most suitable design pattern to address a given set of requirements. (Apply)

Mapping with Programme Outcomes

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	S	S	S	M		M	M	M	S	S	L	S
CO2	S	M	S	M	M	M	M	M	S	S		S
CO3	S	M	S	M	S	M	M	M	S	S		S
CO4	S	S	S	M	S	M	M	M	S	S	L	S
CO5	S	S	S	M	M	M	M	M	S	S	L	S

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	Theory
Remember	-	-	-	-
Understand	20	20	20	20
Apply	30	30	30	80
Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is Factory pattern?
2. Distinguish static class and a singleton class.
3. Give an example where Interpreter pattern is used?

Course Outcome 2 (CO2):

1. Prepare the atomic irreducible requirements for an E-Voting System.
2. Prepare the use case scenario 'System assisted reservation' for a Train Reservation system.
3. Prepare a list of entities relevant to a piece of software concerned with the subject matter Online Exam System. Test and Assess the entities and prepare a refined list of entities and their attributes

Course Outcome 3 (CO3):

1. Analyze and prepare the refined list of entities, their attributes and relationships for an Online Quiz .
2. Design the object types and their interfaces taking into consideration an use case scenario of your choice.
3. Design a set of concrete classes, their instance variables and methods for a E-Voting system.

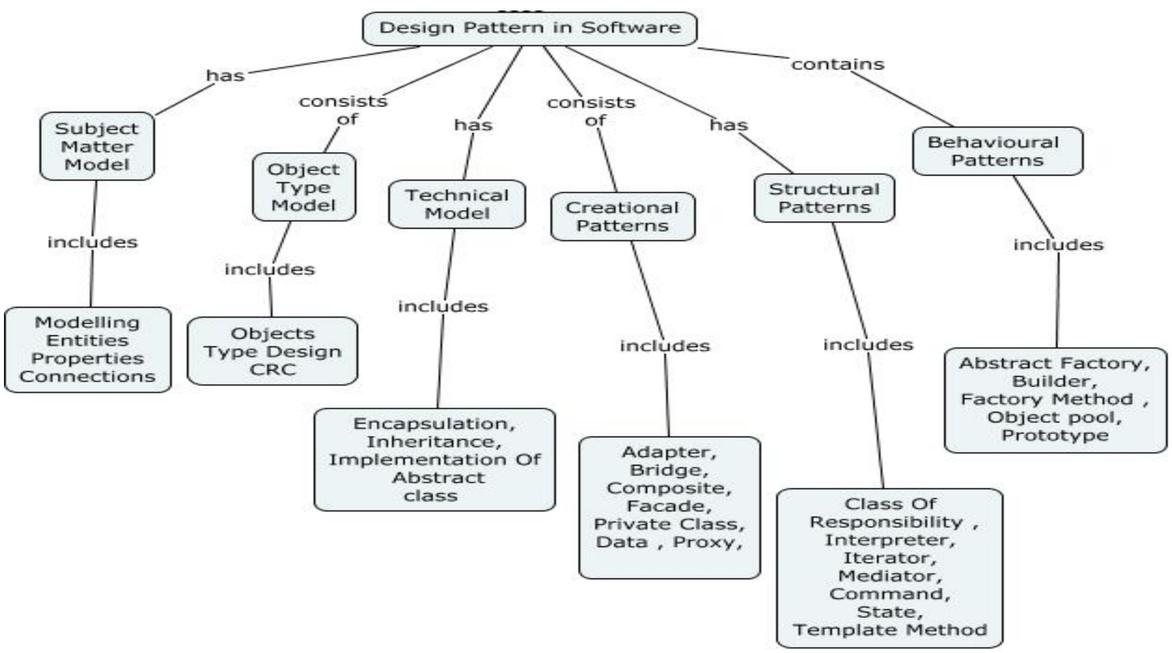
Course Outcome 4 (CO4)

1. Demonstrate the bridge pattern in java by making use of same abstract class.
2. Consider a business case of fast-food restaurant where a typical meal could be a burger and a cold drink. Burger could be either a Veg Burger or Chicken Burger and will be packed by a

wrapper. Cold drink could be either a coke or pepsi and will be packed in a bottle. Demonstrate the builder pattern for the above system.

3. Mix the patterns decorator and bridge for the above fast-food restaurant system and write the consequences.

Concept Map



Syllabus

Introduction – Object and object orientation –Need for analysis and design – Difference and boundary between analysis and design- The Micro development process – The Macro Development process.

Three models – Subject matter model – Object type model – Technical model.

Subject matter model – Modelling – Entities – Properties and connections.

Object Type model – Objects – Type Design – CRC. **Technical Model –** Inheritance – Encapsulation - Relationships - Implementation inheritance and abstract classes

Design Patterns: Creational Patterns – Abstract Factory- Builder – Factory Method – Object Pool – Prototype – Singleton.

Structural Patterns – Adapter- Bridge- Composite- Decorator – Facade – Flyweight – Private Class Data – Proxy

Behavioural Patterns - Chain of responsibility – Command – Interpreter - Iterator – Mediator – Memento – Null Object – Observer - State – Strategy – Template Method – Visitor

Design patterns in cloud, Business process management and Automation testing.

Text Book

1. Eric Freeman & Elisabeth Robson: Head First Design Patterns, O'REILLY, 2014

Reference Books

1. Grady Booch, Robert A.Maksimchuk, Michael W.Engel, Bobbi J.Young, Jim Conallen, Kelli A. Houston, Object-Oriented Analysis and Design with Applications, Third Edition, Addison-Wesley, 2011.
2. John Deacon, Object Oriented Analysis and Design, First Edition, Addison Wesley, 2005.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction (3)**	
1.1	Object and object orientation, Need for analysis and design, Difference and boundary between analysis and design	1
1.2	The Micro development process – The Macro Development process	1
1.3	Three models – Subject matter model – Object type model – Technical model	1
2	**Subject matter model (3)**	
2.1	Modelling	1
2.2	Entities	1
2.3	Properties and connections.	1
3	**Object Type model (4)**	
3.1	Objects	1
3.2	Type Design	1
3.3	CRC	2
4	**Technical Model (4)**	
4.1	Inheritance, Encapsulation	1
4.2	Relationships	1
4.3	Implementation inheritance and abstract classes	2
5	**Design Patterns: Creational Patterns (3)**	
5.1	Abstract Factory, Builder	1
5.2	Factory Method ,Object Pool	1
5.3	Prototype , Singleton	1

Module No.	Topic	No. of Lectures
6	**Structural Patterns (8)**	
6.1	Adapter, Bridge	2
6.2	Composite , Decorator	2
6.3	Facade, Flyweight	2
6.4	Private Class Data – Proxy	2
7	**Behavioural Patterns (8)**	
7.1	Chain of responsibility, Observer	2
7.2	Interpreter , Iterator, Mediator	2
7.3	Command ,Memento ,Null Object	2
7.4	State ,Strategy ,Template Method	2
8	**Case Studies(3)**	
8.1	Design patterns in cloud	1
8.2	Design patterns in Business process management and Automation testing	2
	Total	36

Course Designers:

1. Ms. A.Malini amcse@tce.edu
2. Mr. V. Vignaraj Ananth vignaraj@tce.edu

14CS1A0	INTRODUCTION TO MAINFRAME SYSTEMS	Category	L	T	P	Credit
		OC	1	0	0	1

Preamble

This course is helpful in understanding the basic components required in Enterprise level computing required by large organizations. It explains the components and its features which are essential in for large IT organizations. Also this course helps students to understand how IT department of an organization manages the business needs and expectations from strategy to implementation and daily management of IT infrastructure.

Prerequisite

Nil

Course Outcomes

On the successful completion of the course, students will be able to

Explain the basic concepts of the mainframe, including its use and architecture. (CO1)	Understand
Demonstrate an ability to identify, formulate and solve engineering problems.(CO2)	Apply
Design and conduct experiments, analyze and interpret data.(CO3)	Analyze
Demonstrate an ability to design a system, component or process (CO4)	Create

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S											
CO2.		M		S								
CO3			S									
CO4					S							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	10
Understand	30
Apply	60
Analyse	0
Create	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. How can Linux® on IBM System z® help transform to smarter computing?
2. What are the potential savings of when using Linux on IBM System z?
3. Describe the Characteristics of mainframes.
4. What are the different components of System z?
5. Write about the connectivity between WAS and CICS.

Course Outcome 2 (CO2):

1. Describe the requirements of System z in BFSI, aviation and other sectors.
2. Outline the Overview of ISPF panels.
3. Apply CICS regions for an application.
4. Explain the features of DB2.

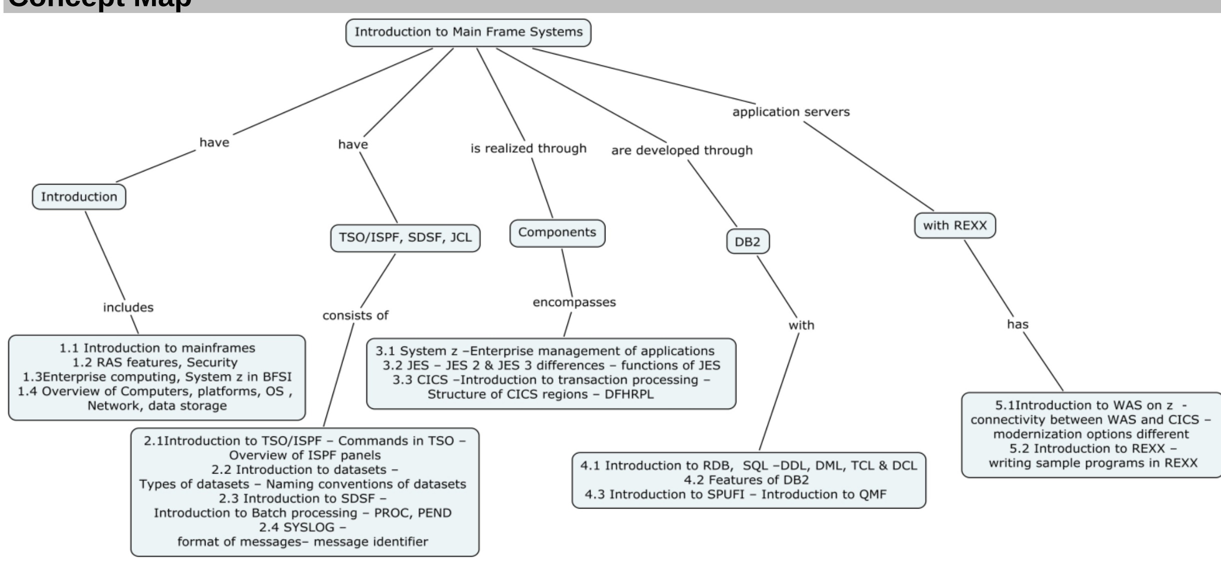
Course Outcome 3 (CO3):

1. While creating a table, by mistake you have given size of one field as 10. But as per requirement size should be 8. What is your next step?
2. File1 has 100 records and file2 has 200 records i want to copy 50 records which r in both file into file3?
3. Want to delete a record from a file in CICS while the region is still up and running. Can you do it with FILE AID and/or CECI?

Course Outcome 4 (CO4):

1. Use TSO Commands.
   a) Create An User ID can have maximum of 8 characters.
   b) To search for a particular word in a text inside a dataset
2. Illustrate JCL Commands.
3. Describe REXX and write a sample program using REXX.

Concept Map



Syllabus

Introduction to System z - Introduction to mainframes- RAS features, Security of mainframes-Enterprise computing, System z in BFSI-Overview of Computers, platforms, OS , Network, data storage

TSO/ISPF, SDSF, JCL - Introduction to TSO/ISPF – Commands in TSO – Overview of ISPF panels- Introduction to datasets –Types of datasets – Naming conventions of datasets-Introduction to SDSF – Introduction to Batch processing – PROC, PEND- SYSLOG –format of messages– message identifier

COMPONENTS of System z- System z –Enterprise management of applications- JES – JES 2 & JES 3 differences – functions of JES- CICS –Introduction to transaction processing –Structure of CICS regions – DFHRPL

DB2- Introduction to RDB, SQL –DDL, DML, TCL & DCL- Features of DB2- Introduction to SPUFI – Introduction to QMF

WAS - Introduction to WAS on z - connectivity between WAS and CICS – modernization options different - REXX – Introduction to REXX –writing sample programs in REXX

Reference Books

z/OS Basic Skills Information Center
http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp - contains 30min video presentations
http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp?topic=/com.ibm.zos.zjcl/zjclt_howto_usecoll.htm

IBM Education Assistant
http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp

z/OS Internet Library
http://www.ibm.com/systems/z/os/zos/bkserv/

System z Home Page
http://www.ibm.com/systems/z/index.html

Collection of z/OS URLs
http://www.ibm.com/systems/z/os/zos/zfavorites/

IBM Redbooks (How-To Books)
http://www.redbooks.ibm.com/ - Search word 'ABCs' returns 12 volumes on major z/OS topics

System z - Academic Initiative program
https://www.ibm.com/developerworks/university/systemz/ - **For University Professors and Students**

System and Product Messages - LookAt
http://www-03.ibm.com/systems/z/os/zos/bkserv/lookat/index.html - **Internet IBM z/OS Messages and Codes**

CICS Information Centers
http://www-01.ibm.com/software/htp/cics/library/indexes/infocenters.html

DB2 for z/OS - Technical Resources
http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011656

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction to System z**	
1.1	Introduction to mainframes	1

1.2	RAS features, Security of mainframes	1
1.3	Enterprise computing , System z in BFSI	1
1.4	Overview of Computers, platforms, OS , Network, data storage	1
2	**TSO/ISPF, SDSF, JCL**	
2.1	Introduction to TSO/ISPF – Commands in TSO – Overview of ISPF panels	1
2.2	Introduction to datasets –Types of datasets – Naming conventions of datasets	1
2.3	Introduction to SDSF – Introduction to Batch processing – PROC, PEND	1
2.4	**SYSLOG –**format of messages– message identifier	1
3	**COMPONENTS of System z**	
3.1	System z –Enterprise management of applications	1
3.2	**JES** – JES 2 & JES 3 differences – functions of JES	1
3.3	**CICS** –Introduction to transaction processing – Structure of CICS regions – DFHRPL	1
4	**DB2**	
4.1	Introduction to RDB, SQL –DDL, DML, TCL & DCL	1
4.2	Features of DB2	1
4.3	Introduction to SPUFI – Introduction to QMF	1
5	**WAS**	
5.1	Introduction to WAS on z - connectivity between WAS and CICS – modernization options different	1
5.2	**REXX** – Introduction to REXX –writing sample programs in REXX	1
	Total	**16**

Course Designers:

1. Mr. K.Shreekanth kshreeka@in.ibm.com
2. Mr. V.Vignaraj Ananth vignaraj@tce.edu

14CS1B0	MOBILE APPLICATION DEVELOPMENT	Category	L	T	P	Credit
		OC	1	0	0	1

Preamble

This course will provide specialized knowledge on computing with focus on mobile applications technology. Students will be trained in understanding the concepts of emerging technologies in mobile computing and development of applications to be run on mobile devices. Students will study and gain experience with the languages and frameworks that are most commonly used in developing these applications, with the design of user interfaces and software systems, and with associated topics such as networking, hosting infrastructure, and security. Students will also learn the fundamental principles on which these topics are based, so that they will be prepared for the new technologies that are constantly being developed.

Prerequisite

- 14CS270 Problem Solving using Computers
- 14CS370 Object Oriented Programming

Course Outcomes

Describe the major mobile device platforms and their capabilities along with leading technologies for mobile app development. (CO1)	Understand
Illustrate the Mobile Application architecture. (CO2)	Understand
Use the development environments and languages to simulate modules of mobile application like form validation, navigation, etc. (CO3)	Apply
Develop small mobile apps with given specification like adding REST services, attaching note to photo, etc. (CO4)	Create*

*Course Outcome 4 (CO4) is validated through Assignments

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M				S							
CO2	S				M							
CO3	M	M	S	L	M	L	L	L				M
CO4	M	S	S	M	M	M	M	S	S	S	M	M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
	Theory
Remember	0
Understand	50
Apply	50
Analyse	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List the various technology options available for development like native development, cross platform development platforms, rich web. (Remember)
2. Report the device capabilities of Android and iOS based smart phones and Tablets. (Understand)

Course Outcome 2 (CO2):

1. Describe the Apple (iOS) and Android Ecosystem. (Understand)
2. Discuss the enterprise integration technologies. (Understand)

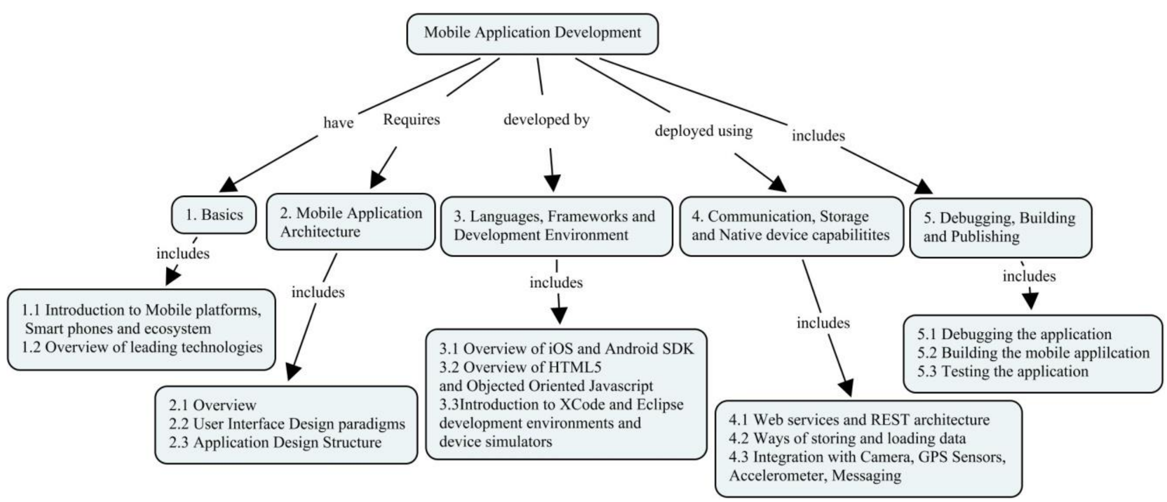
Course Outcome 3 (CO3):

1. Practice XCode IDE on Apple Mac Desktop. (Apply)
2. Use the online guides from Apple and Google on iOS and Android SDK. (Apply)

Course Outcome 4 (CO4):

1. Develop a REST service to receive data from the notes mobile application and save it on a server database. (Create)
2. Enhance the notes mobile application to communicate with the server over REST protocol and save the notes on the server. (Create)

Concept Map



Syllabus

Basic Concepts: Introduction to mobile platforms, smart phones and ecosystem. Overview of leading technologies available for mobile app development. **Mobile Application Architecture:** Overview of mobile application architecture – iOS and Android. Mobile user interface design paradigms – dynamic interfaces, multi-touch gestures, animations Application Design Structure. **Languages, Frameworks and Development Environments:** Overview of iOS and Android SDK, Overview of HTML5 and Objected Oriented Javascript, Introduction to XCode and Eclipse development environments and device simulators. **Communication, Storage and Native Device Capabilities:** Overview of web services and REST architecture, Multiple ways of storing and loading data, Integration with Camera, GPS Sensors, Accelerometer, Messaging. **Debugging, Building and Publishing:** Debugging the app on simulator and actual mobile device, Building the mobile app and publishing to the market, Testing the mobile app.

References

1. Joe Conway, Aaron Hillegass, "iPhone Programming: The Big Nerd Ranch Guide", 2nd Edition, Addison-Wesley Professional, ISBN 978-0321773777.
2. Wei-Meng-Lee, "Beginning Android Application Development", 1st Editon, Wrox, ISBN 978-1118017111.
3. Jonathan Stark, "Building iPhone Apps with HTML, CSS, and JavaScript: Making App Store Apps Without Objective-C or Cocoa", 1st Edition, O'Reilly Media, ISBN 978-0596805784.
4. Jonathan Stark, "Building Android Apps with HTML, CSS, and JavaScript", 1st Edition, O'Reilly Media, ISBN 978-1449383268.
5. iOS Developer's Library - http://developer.apple.com/library/ios/navigation/
6. Android Developer's Guide - http://developer.android.com/guide/index.html

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Basic Concepts**	
1.1	Introduction to mobile platforms, smart phones and ecosystem	1
1.2	Overview of leading technologies available for mobile development	1
2	**Mobile Application Architecture**	
2.1	Overview of mobile application architecture – iOS and Android	1
2.2	Mobile user interface design paradigms – dynamic interfaces, multi-touch gestures, animations	1
2.3	Application Design Structure	1
3	**Languages, Frameworks and Development Environments**	
3.1	Overview of iOS and Android SDK	1
3.2	Overview of HTML5 and Objected Oriented Javascript	1
3.3	Introduction to XCode and Eclipse development environments and device simulators	1
4	**Communication, Storage and Native Device Capabilities**	

4.1	Overview of web services and REST architecture	1
4.2	Multiple ways of storing and loading data	1
4.3	Integration with Camera, GPS Sensors, Accelerometer, Messaging	1
5	**Debugging, Building and Publishing**	
5.1	Debugging the app on simulator and actual mobile device	1
5.2	Building the mobile app and publishing to the market	1
5.3	Testing the mobile app	1
	Total No of Hours	14

Course Designers:

1. M. Suguna mscse@tce.edu
2. Gangadhar Neeli Gangadhar.Neeli@honeywell.com
3. Subramanian Krishnan subramanian.krishnan@honeywell.com

14CS1C0	PRACTICAL APPROACHES TO NETWORKING	Category	L	T	P	Credit
		OC	1	0	0	1

Preamble

The course takes one step further to the networking fundamentals already learnt by the student through the "Computer Networks" theory course in the curriculum. This course assists the student to understand and familiarize with the practical and managerial aspects of several common Networking Services like DNS, DHCP etc., through hands on labs and some packet analyzing tools.

Prerequisite

Basics of Computer Networks

Course Outcomes

On the successful completion of the course, students will be able to

Explain the concept of RPC (CO1)	Understand
Get familiar with using network packet analysis tools, to understand, identify and troubleshoot network related issues (CO2)	Apply
Demonstrate a deep understanding of the common networking services like DNS and DHCP (CO3)	Apply
Implementation of common networking services and Virtual machine in the lab (C04)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	M										
CO2.	S	S										
CO3.	S	S										
CO4.	S	S										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	20
Understand	30
Apply	50
Analyse	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define RPC.(Remember)
2. State the difference between LPC and RPC. (Understand)

Course Outcome 2 (CO2):

1. Define DORA.(Remember)
2. Distinguish between TCP and UDP. (Understand)
3. Capture one sequence of TCP/IP Handshake using NETMON.(Apply)

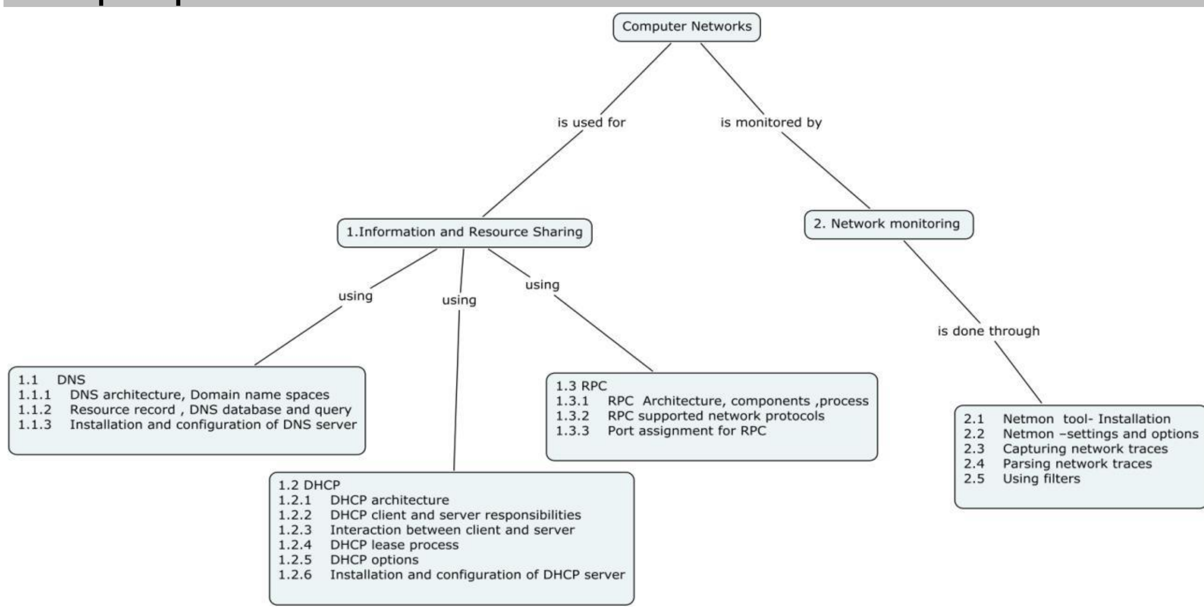
Course Outcome 3 (CO3):

1. State the need for DNS and DHCP. (Remember)
2. Explain DHCP lease process. (Understand)
3. Install and implement DHCP server in your network (Apply)

Course Outcome 4 (CO4):

1. State the command used to identify the IP address of local machine.(Remember)
2. Explain the need for private network. (Understand)
3. Create two Virtual machines using hyper-V. Install a server and a client operating system in respective machines. Do manual IP configuration to client machine. And establish private network between client and server. (Apply)

Concept Map



Syllabus

Information and Resource sharing: DNS, DNS architecture, Domain name spaces, Resource record , DNS database and query, Installation and configuration of DNS server, **DHCP**, DHCP architecture ,DHCP client and server responsibilities, Interaction between client and server, DHCP lease process, DHCP options, Installation and configuration of DHCP server, **RPC**, RPC Architecture, components ,process, RPC supported network protocols, Port assignment for RPC **Network monitoring**- Netmon tool- Installation, settings and options, Capturing network traces, Parsing network traces, Using filters**.**

References

1. http://technet.microsoft.com/en-us/library/cc787921(v=ws.10).aspx – What is DNS?
2. http://technet.microsoft.com/en-us/library/cc772774(v=ws.10).aspx – How DNS Works

3. http://technet.microsoft.com/en-us/library/cc775464(v=ws.10).aspx – DNS Tools and Settings
4. http://technet.microsoft.com/en-us/library/cc725925.aspx - Install a DNS Server in Windows Server 2008 R2
5. http://technet.microsoft.com/en-us/library/cc771031.aspx - Configure a new DNS Server
6. http://technet.microsoft.com/en-us/library/cc781008(v=ws.10).aspx – What is DHCP?
7. http://technet.microsoft.com/en-us/library/cc780760(v=ws.10).aspx – How DHCP Technology Works
8. http://technet.microsoft.com/en-us/library/cc782411(v=ws.10).aspx – DHCP Tools and Settings
9. http://technet.microsoft.com/en-us/library/cc732075.aspx - Installing DHCP Server Role
10. http://technet.microsoft.com/en-us/library/cc732584.aspx - Configuring DHCP Server Role Settings
11. http://technet.microsoft.com/en-us/library/cc787851(v=ws.10).aspx – What is RPC?
12. http://technet.microsoft.com/en-us/library/cc738291(v=ws.10).aspx – How RPC Works
13. How to use Network Monitor to capture network traffic -- http://support.microsoft. com/ kb/812953
14. Frequently Asked Questions About Network Monitor -- http://support.microsoft.com/ kb/294818
15. The Basics of Reading TCP/IP Traces -- http://support.microsoft.com/kb/169292
16. Explanation of the Three-way Handshake via TCP/IP - http://support.microsoft.com/ kb/172983
17. Blog - http://blogs.technet.com/b/netmon/
18. http://channel9.msdn.com/tags/Netmon/- Channel9 Netmon Videos

Course Contents and Lecture Schedule

No.	Topics	No of Lectures
1	**Information and Resource Sharing**	
1.1	**DNS**	
1.1.1	DNS architecture, Domain name spaces	1
1.1.2	Resource record , DNS database and query	1
1.1.3	Installation and configuration of DNS server	2
1.2	**DHCP**	
1.2.1	DHCP architecture	1
1.2.2	DHCP client and server responsibilities	
1.2.3	Interaction between client and server	
1.2.4	DHCP lease process	1
1.2.5	DHCP options	
1.2.6	Installation and configuration of DHCP server	2
1.3	RPC	
1.3.1	RPC Architecture, components ,process	2
1.3.2	RPC supported network protocols	
1.3.3	Port assignment for RPC	
2	**Network monitoring**	
2.1	Netmon tool- Installation	1

2.2	Netmon –settings and options	
2.3	Capturing network traces	1
2.4	Parsing network traces	1
2.5	Using filters	1
	Total	**14**

Course Designers:

1. Mr. C.Senthilkumar cskcse@tce.edu
2. Mr. T.Manikandan tmcse@tce.edu

		Category	L	T	P	Credit
14CS1D0	**EMBEDDED SOFTWARE DEVELOPMENT**	OC	1	0	0	1

Preamble

The course discusses the basic programming concepts required for the embedded software development. It includes the concept of process, thread, memory management and interfacing concepts covered in both assembly language and c programming.

Prerequisite

14CS240 - Computer Organisation and Microprocessors

Course Outcomes

On the successful completion of the course, students will be able to

Explain the types and categories of Real time systems and programming concepts. (CO1)	Understand
Make use of the effect of cache on embedded software. (CO2)	Apply
Demonstrate the concepts of Spinlock, semaphore and ISR. (CO3)	Apply
Summarize the optimized coding techniques. (C04)	Understand

Mapping with Programme Outcomes

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1.	M	M										
CO2.	M	S										
CO3.	M	S										
CO4.	M	M										

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	20
Understand	20
Apply	60
Analyse	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

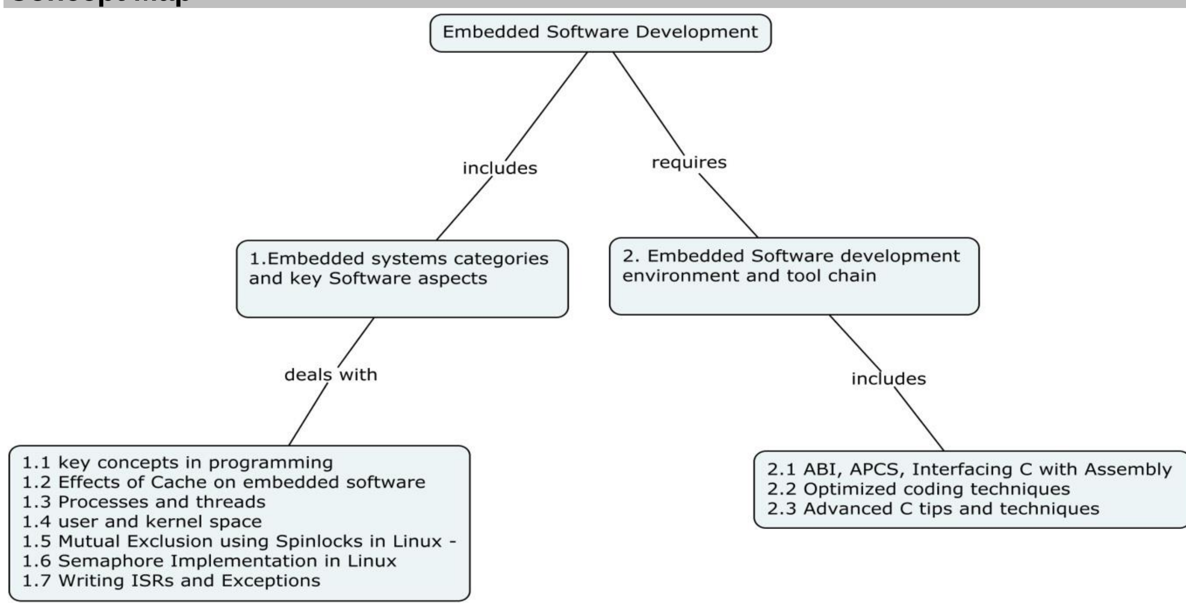
1. Define Real time systems.(Remember)
2. State the different categories of real time systems.(Remember)
3. Distinguish between process and thread. (Understand)

Course Outcome 2 (CO2):
1. List the different cache memory organization.(Remember)
2. Distinguish between write through and write cache. (Understand)
3. Identify the need for multi level cache in embedded system environment.(Apply)

Course Outcome 3 (CO3):
1. State the need for semaphore. (Remember)
2. Explain the ISR process. (Understand)
3. Apply the concept of semaphore to alleviate the critical section problem. (Apply)

Course Outcome 4 (CO4):
1. List the different code optimization techniques.(Remember)
2. Explain the need for profiler. (Understand)
3. Explain the ARM software development flow. (Understand)

Concept Map



Syllabus

Embedded systems categories and key Software aspects –key concepts in programming – Effects of Cache on embedded software - Processes and threads, user and kernel space – Mutual Exclusion using Spinlocks in Linux - Semaphore Implementation in Linux – Writing ISRs and Exceptions – **Embedded Software development environment and tool chain** – ABI, APCS, Interfacing C with Assembly – Optimized coding techniques – Advanced C tips and techniques.

References

1. Jane W.S Liu, “Real time systems”, Pearson Education, 2000.
2. Daniel P. Bovet, Marco Cesati, ‘Understanding the Linux Kernel’,O’REILLY ,Third edition, 2006

3. Peter Barry, Patrick Crowliey,’ Modern Embedded Computing: Designing Connected, Pervasive, Media-rich System”, Elsevier, 2012.

4. Andrew Koenig,” C Traps and Pitfalls”, Pearson Education, 2007.

Course Contents and Lecture Schedule

No.	Topics	No of Lectures
1	**Embedded systems categories and key Software aspects**	
1.1	key concepts in programming	2
1.2	Effects of Cache on embedded software	1
1.3	Processes and threads	1
1.4	User and kernel space	1
1.5	Mutual Exclusion using Spinlocks in Linux	1
1.6	Semaphore Implementation in Linux	2
1.7	Writing ISRs and Exceptions	1
2	**Embedded Software development environment and tool chain**	
2.1	ABI, APCS, Interfacing C with Assembly	2
2.2	Optimized coding techniques	2
2.3	Advanced C tips and techniques	1
	Total	**14**

Course Designers:

1. Mr.Mouli Sankaran mouli.sankaran@yahoo.com
2. Dr.C.Senthilkumar cskcse@tce.edu
3. Mr.T.Manikandan tmcse@tce.edu

14CS1E0	INTRODUCTION TO IT SERVICE MANAGEMENT	Category	L	T	P	Credit
		PE	1	0	0	1

Preamble

This course is helpful for students who are keen on getting the basic understanding of the ITIL framework and on how it can be used to enhance the quality of IT services within an organization. Also this course helps students to understand how IT department of an organization manages the business needs and expectations from strategy to implementation and daily management of IT as a service. Also this course will help the students in understanding the linkage between ITIL and cloud computing.

Course Outcomes

Show the impact of engineering solutions on the society and be aware of contemporary issues. (CO1)	Remember
Demonstrate an ability to visualize and work on laboratory and multi-disciplinary tasks. (CO2)	Understand
Use of modern engineering tools, software and equipment to analyze problems. (CO3)	Apply
Develop a system, component or process as per needs and specifications. (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M				S							
CO2	S				M							
CO3	M	M	S	L	M	L	L	L				M
CO4	M	S	S	M	M	M	M	S	S	S	M	M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	10
Understand	30
Apply	60
Analyze	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Recall about RFID wireless network.
2. Write about Responsiveness of a service
3. Identify various components in ITIL lifecycle
4. Describe about Service Portfolio Design
5. Write about QoS.

Course Outcome 2 (CO2):

1. Demonstrate proactive risk strategy with one example.
2. Exhibit the steps involved in service design.

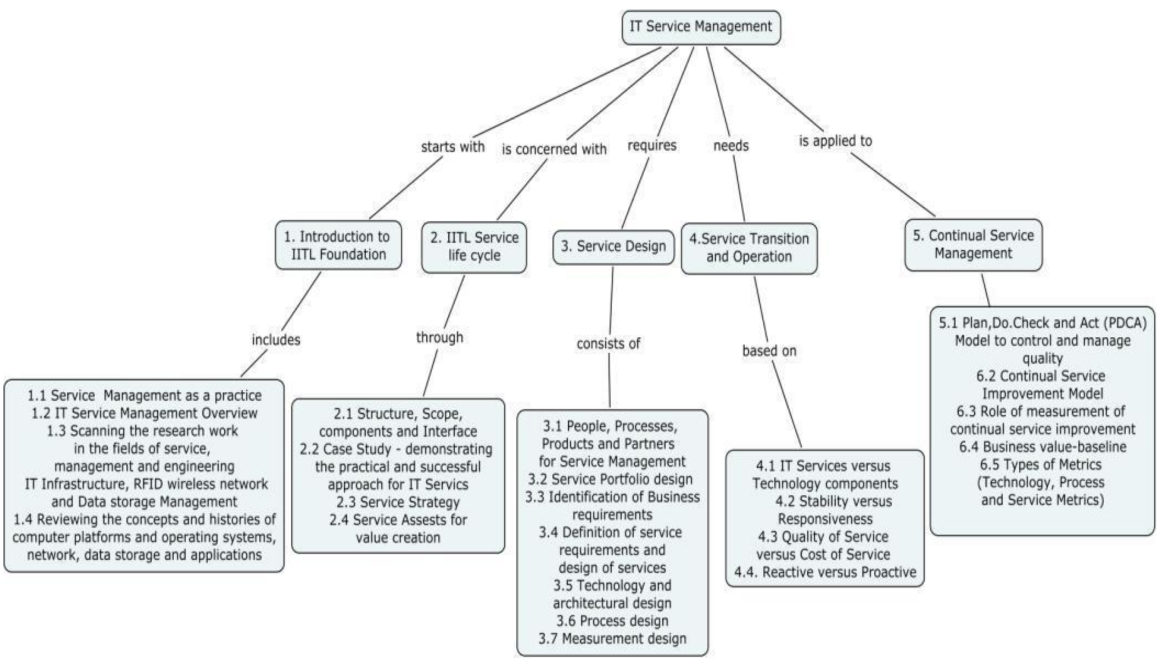
Course Outcome 3 (CO3):

1. Illustrate PDCA Model to control and manage quality.
2. Apply ITIL service lifecycle for any software application.

Course Outcome 4 (CO4):

1. Develop a measurement service design for android application.
2. Apply the process and technology metric for Motorola’s program.

Concept Map



Syllabus

Introduction to ITIL Foundation - Service Management as a Practice - IT Service Management Overview - scanning the research work in the fields of service science, management, and engineering IT Infrastructure, RFID wireless network, and Data Storage Management - reviewing the concepts and histories of computer platforms and operating systems, network, data storage, and applications.**ITIL Service Lifecycle** - Structure, Scope, Components and Interface. Case study – demonstrating the practical and successful approaches for IT services. Service Strategy - Service Assets are the basis for Value Creation.**Service Design** - People, Processes, Products and Partners for Service Management, Service Portfolio Design, Identification of Business Requirements, definition of Service requirements and design of Services, Technology and architectural design, Process design, Measurement design.**Service Transition and Operation** – IT Services versus Technology components, Stability versus Responsiveness, Quality of Service versus Cost of Service, and Reactive versus Proactive.**Continual Service Improvement** - Plan, Do, Check and Act (PDCA) Model to control and manage quality, Continual Service Improvement Model, The role of measurement for Continual Service Improvement, Business value- Baseline, Types of metrics (technology metrics, process metrics, service metrics).

References

1. http://www.best-management-practice.com/IT-Service-Management-ITIL/?ClickID=004798
2. The Introduction to the ITIL Service Lifecycle Book (Paperback), Office of Government Commerce, Published, 2007, The Stationary Office.
3. Service Management, Fourth Edition, J.A. Fitzsimmons and M.J. Fitzsimmons, McGraw Hill
4. http://www-935.ibm.com/services/us/en/it-services/it-service-management-implementation.html
5. http://www-935.ibm.com/services/be/en/it-services/it-management-consulting-services.html

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction to ITIL Foundation**	
1.1	Service Management as a Practice	1
1.2	IT Service Management Overview	1
1.3	Scanning the research work in the fields of service science, management and engineering IT Infrastructure, RFID wireless network and Data Storage Management	1
1.4	Reviewing the concepts and histories of computer platforms and operating systems, network, data storage, and applications	1
2	**ITIL Service Lifecycle**	
2.1	Structure, Scope, Components and Interface.	1

2.2	Case study – demonstrating the practical and successful approaches for IT services	1
2.3	Service Strategy - Service Assets are the basis for Value Creation	1
3	**Service Design**	
3.1	People, Processes, Products and Partners for Service Management	1
3.2	Service Portfolio Design, Identification of Business Requirements, definition of Service requirements and design of Services	1
3.3	Technology and architectural design, Process Design, Measurement Design.	1
4	**Service Transition and Operation**	
4.1	IT Services versus Technology components, Stability versus Responsiveness.	1
4.2	Quality of Service versus Cost of Service, Reactive versus Proactive	1
5	**Continual Service Improvement**	
5.1	Plan, Do, Check and Act (PDCA) Model to control and manage quality	1
5.2	Continual Service Improvement Model, The role of measurement for Continual Service Improvement	1
5.3	Business value- Baseline	1
5.4	Types of metrics (technology metrics, process metrics, service metrics)	1
	Total	**16**

Course Designers:

1. Dr. Sampath Na Parthasarathy- sambath.narayanan@in.ibm.com

2. Mr.V.Vignaraj Ananth - vignaraj@tce.edu

		Category	L	T	P	Credit
14CS1F0	**GREEN DATA CENTER**	PE	1	0	0	1

Preamble

The objective of this course is to introduce the concept of Green Datacenter and the important role it plays in modern day computing. This Green Data Center course teaches a comprehensive technical challenges and solutions for rapidly growing challenges and trends in Green Data Center through various corporate datacenters. It is designed to kindle interest for the student to explore the exciting world of Green IT and the interaction between the world of software and hardware designs in Energy Efficient Green Data Center

Prerequisite

- Basic Knowledge in computer architecture and computing

Course Outcomes

On the successful completion of the course, students will be able to

Explain the components of data centre, problems faced by data center and the need for energy efficient Green Data center (CO1)	Understand
Interpret the importance of virtualization for Green data center (CO2)	Understand
Identify an appropriate cooling technologies and infrastructure for optimizing the cost of data center operations (CO3)	Apply
Evaluate the ways to achieve Green Data center goals and future Technologies in DC (CO4)	Evaluate

Mapping with Programme Outcomes

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M	L				M	M					L
CO2.	M	L				L	L					L
CO3.	S	M	M	L		M	M					L
CO4.	S	S	S	M		M	M					L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	0
Understand	30
Apply	50
Analyse	0
Evaluate	20
Create	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the need for Datacenter.
2. List out the problems faced by data center.
3. Report the ways to reduce active power in the CPU at run time.
4. Discuss the different ways to reduce the power in memory and I/O.
5. What is the need for energy efficient data center?

Course Outcome 2 (CO2):

1. Classify the types of virtualization.
2. Associate the impact of virtualization to Green Data Center.
3. Describe the importance of software defined data center.
4. Outline the significance of cloud computing.
5. Explain the concept of consolidation.

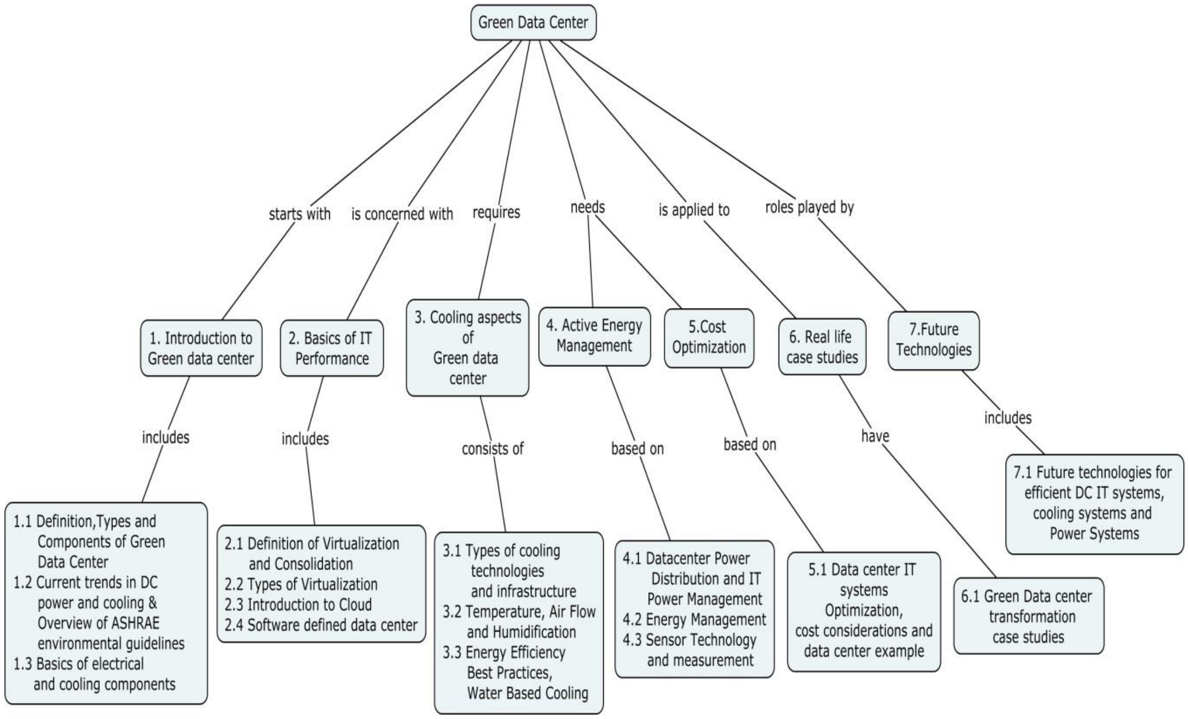
Course Outcome 3 (CO3)

1. Apply ASHRAE environment guidelines to maintain temperature and humidity for the IT systems in a datacenter.
2. Identify the ways of optimizing the IT Energy Consumption
3. Choose the appropriate techniques for cooling the server in a data center
4. Identify the suitable active energy management technique to maintain a data center.
5. Make use of various measurement to maintain energy efficient data center.

Course Outcome 4 (CO4)

1. Evaluate the ways to monitor and manage the server/IT system power consumption using management tools.
2. Evaluate the ways to achieve Green Data Centre goals.
3. Determine the ways to overcome the problems faced by a data center day to day.
4. Evaluate best method of cooling a 30 rack.
5. Predict any three companies that have build datacenters and follow energy efficiency practices with neat sketch

Concept Map



Syllabus

Introduction to Green Datacenter: Typical Definition of DC, need for a datacenter, various components in a datacenter,current trends in DC power and cooling, overview of ASHRAE environmental guidelines, Basics of electrical and cooling components in data center. **Basics of IT Performance:** Definition of Virtualization and Consolidation, Types of Virtualization, Introduction to Cloud, Software defined data center. **Cooling Aspects of Green Datacenter**: Overview of different types of cooling technologies and infrastructure for a data center, Temperature, Airflow and Humidification in a data center, Energy efficiency Best Practices, water based cooling. **Active Energy Management:** Power distribution systems in a datacenter like UPS system, PDU systems and ways to monitor and manage the power distribution systems, Energy management, Sensor technology and measurement in Data center. **Cost optimization:** Data center IT Systems Optimizations, Cost considerations. **Real Life Case Studies:** Videos and presentation on Green Datacenter transformation case studies. **Future Technologies:** Future technologies for efficient DC IT systems, Cooling systems like free cooling, direct water cooling to servers and datacenter infrastructure.

Reference Books

1. Bernard Golden, “Virtualization For Dummies”, John Wiley & Sons,2007.
2. Victor Moreno, “Network Virtualization”, Kindle Edition, CISCO Press.

3. Tom Clark, " Storage Virtualization: Technologies for Simplifying Data Storage and Management", First Edition,2005, ISBN-10: 0321262514

4. Database virtualization : sg247805. pdf at http://www.ibm.com/redbooks

5. White Paper: "Impact of Virtualization on Datacenter" BY Dennis Boule

6. Linux Power Management: http://www.ruf.rice.edu/~mobile/elec518/lectures/2011-tatepeter.pdf

7. Mickey Iqbal ,Mithkal Smadi , Chris Molloy , and Jim Rymarczyk, "IT Virtualization Best Practices: A Lean, Green Virtualized Data Center Approach", 2011, MC Press, Ketchum, ID 83340

8. Evolution of Data Center Environmental Guidelines, Roger R Schmidt et. al., ASHRAE Transactions

9. http://www.thegreengrid.org

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction to Green Datacenter(3)**	
1.1	Definition of Green Data Center & Types and Components of Data center	1
1.2	Current trends in DC power and cooling, Overview of ASHRAE environmental guidelines	1
1.3	Basics of electrical and cooling components in data center	1
2	**Basics of IT Performance(Virtualization, Cloud, Software Defined Data Center) (4)**	
2.1	Definition of Virtualization and Consolidation	1
2.2	Types of Virtualization	1
2.3	Introduction to Cloud	1
2.4	Software defined data center	1
3	**Cooling Aspects of Green Datacenter (3)**	

3.1	Overview of different types of cooling technologies and infrastructure for a Datacenter	1
3.2	Temperature, Air Flow and Humidification in a Datacenter	1
3.3	Energy Efficiency Best Practices, Water Based Cooling	1
4	**Active Energy Management (3)**	
4.1	Datacenter Power Distribution and IT Power Management	1
4.2	Energy Management	1
4.3	Sensor Technology and measurement in Data centers	1
5	**Cost Optimization (1)**	
5.1	Data center IT Systems Optimizations, Cost Considerations and Data center Example	1
6	**Real life case Studies (1)**	
6.1	Green Data center transformation case studies	1
7	**Future technologies (1)**	
7.1	Future technologies for efficient DC IT systems, cooling systems and Power Systems	1
	Total No. of. hours	**16**

Course Designers:

1. Dr.Sampath Na.Parthasarathy sambath.narayanan@in.ibm.com
2. Mr.Vidhya Shankar vidhya.shankar@in.ibm.com
3. Dr.S.Padmavathi spmcse@tce.edu

14CS1G0	**BASICS OF APPLICATION SECURITY**	Category	L	T	P	Credit
		PE	1	0	0	1

Preamble

The main intention of application security is to enforce an application with sound/effective security routines that minimize the probability of an attacker from being able to manipulate applications and access, steal, modify or delete sensitive data by unauthorised means.

Prerequisite

- 14CS620 - Internet Programming
- 14CS370 - Object Oriented Programming

Course Outcomes

On the successful completion of the course, students will be able to

List the application threats (CO1)	Understand
Understand the concepts of Information and Application security.(CO2)	Understand
Apply the secure coding principles while developing real time applications. (CO3)	Apply
Analyse different types of attacks on web applications (CO4)	Analyse
Rate the security of an application (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L											
CO2	L											
CO3	S		S	L		L	M					L
CO4	M	M			L	L	L					L
CO5	S	M	M		L	L	M	L				L

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	20
Understand	30
Apply	40
Analyse	10
Evaluate	0
Create	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List the common web application protocols.

2. Recall the vulnerabilities in a web application.
3. Which attack can execute scripts in the user's browser and is capable of hijacking user sessions, defacing websites or redirecting the user to malicious sites?
4. What threat arises from not flagging HTTP cookies with tokens as secure?
5. What is address spoofing?

Course Outcome 2 (CO2):

1. Understand the basics of Networking and about the commonly used protocols.
2. Understand different classes of application security and their real world implications.
3. Understand the common attacks on web applications and their countermeasures.
4. Understand the basic concepts of Secure Coding principles.
5. How to secure a database?

Course Outcome 3 (CO3):

1. Illustrate the secure coding guideline with an example.
2. Demonstrate security routines which would act as defensive measures for different web application attacks.
3. Identify the hidden security threats in the given vulerale demo application
4. Construct a secure database for a web application.
5. Contrast the SDLC framework and secure SDLC framework.

Course Outcome 4 (CO4):

1. Compare and understand the secure coding guidelines via a vulnerable code base.
2. Download and explore a vulnerable demo application.
3. Understand and analyze the implications of Secure SDLC process and try to incorporate the same in real time development projects
4. Analyze different infrastructure level configurations and prescribe countermeasures for the security loopholes identified (if any).
5. Analyze how different attacks occur and about how to bypass the currently implemented filters/countermeasures.

Course Outcome 5 (CO5):

1. Critique the web application security in YAHOO MAIL.
2. Identify ad assess potential threats and vulnerabilities of TCENET.

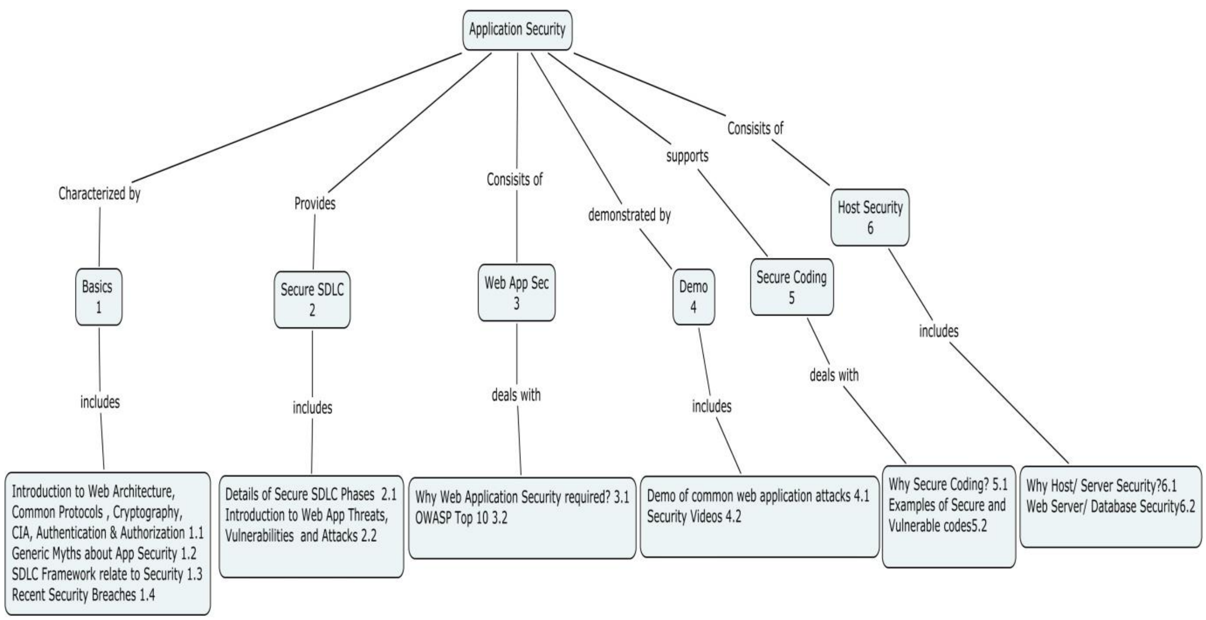
3. Rate the techniques that are best from a security standpoint in handling “Forgot Password”?

4. Estimate the vulnerabilities in the given code.

5. Critique the effect of SQL injection I the following code.

   SELECT booktitle **FROM** booklist **WHERE** bookId = ‘ook14cd’ **AND** ‘1’=’1’

Concept Map



Syllabus

Introduction: Introduction to Web Architecture, Common Protocols (SSL/HTTP), Cryptography, CIA, Authentication & Authorization, Generic Myths about App Security, SDLC Framework relate to Security, Recent Security Breaches. **Secure SDLC:** Details of Secure SDLC Phases (Requirement, Design, Coding, Testing), Introduction to Web App Threats, Vulnerabilities and Attacks. **Web App Sec:** Why Web Application Security required? OWASP Top 10, **Demo:** Demo of common web application attacks, Security Videos. **Secure Coding:** Why Secure Coding? Examples of Secure and Vulnerable codes. **Host Security:** Why Host/ Server Security? Web Server/ Database Security

Text Books

1. Dafydd Stuttard ,Marcus Pinto,”The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws,”2nd Edition,Wiley,2011,ISBN:1118026470/978-1118026472.

2. Stuart McClure, Joel Scambray,Kurtz,” Hacking Exposed 7:Network Security Secrets & Solutions”, 7thEdition, McGraw-Hill Prof Med/Tech, 2012, ISBN13:9780071780285

3. Andrew S. Tanenbaum and David J. Wetherall,”Computer Networks”,5th Edition,Pretice Hall,2011,IB-13:9780132126953

4. OWASP Security Testing Guide (https://www.owasp.org/index.php/File:OWASP_Testing_Guide_v2_pdf.zip)

Web References

OWASP: http://www.owasp.org
WASC: http://www.webappsec.org/
SANS: http://www.sans.org/

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Basics**	
1.1	Introduction to Web Architecture, Common Protocols (SSL/HTTP), Cryptography, CIA, Authentication & Authorization	1
1.2	Generic Myths about App Security	1
1.3	SDLC Framework relate to Security	1
1.4	Recent Security Breaches	1
2	**Secure SDLC**	
2.1	Details of Secure SDLC Phases (Requirement, Design, Coding, Testing)	1
2.2	Introduction to Web App Threats, Vulnerabilities and Attacks	1
3	**Web App Sec**	
3.1	Why Web Application Security required?	1
3.2	OWASP Top 10	2
4	**DEMO**	
4.1	Demo of common web application attacks	1
4.2	Security Videos	1
5	**Secure Coding**	
5.1	Why Secure Coding?	1
5.2	Examples of Secure and Vulnerable codes	1
6	**Host Security**	
6.1	Why Host/ Server Security?	1
6.2	Web Server/ Database Security	1
	Total No. of Hours	**15**

Course Designers:

1. Satheesh PRV — Satheesh.Kumar.P.R.Veerapadran@honeywell.com
2. Ravikanth Dangeti — Ravinkanth.Dangeti@honeywell.com
3. Akash Shrivastava — Akash.Shrivastava@Honeywell.com
4. Ramesh Sitaraman — Ramesh.Sitaraman@Honeywell.com
5. Murali Krishnan — Murali.KrishnanS@Honeywell.com
6. M.Vijayalakshmi — mviji@tce.edu

		Category	L	T	P	Credit
14CS1H0	**FOUNDATIONS OF NOSQL DATABASE**	PE	1	0	0	1

Preamble

This course will provide specialized knowledge on computing with focus on hadoop applications technology. Students will be trained in understanding the concepts of emerging technologies in data analytics and development of applications to be run on hadoop environment. Students will study and gain experience with the languages and frameworks that are most commonly used in developing these applications, with the design of user interfaces and software systems, and with associated topics such as map reduce framework, hosting infrastructure, and database administration. Students will also learn the fundamental principles on which these topics are based, so that they will be prepared for the new technologies that are constantly being developed.

Objectives:

The Student will be able to:

- Explain the elements of NoSQL design
- Set up NoSQL Database
- Create and execute Pig Scripts
- Install and configure Hive
- Execute HSQL for data management

Prerequisite

- 14CS370 Object Oriented Programming
- 14CS440 Database Management System

Course Outcomes

Describe the relational databases and data models in NoSQL (CO1)	Understand
Illustrate way to run pig programs in the editor for data processing (CO2)	Understand
Develop the given application using NoSQL based technologies on top of Hadoop and identify its merits over traditional database implementations (CO3)	Apply
Develop Map Reduce script using HiveQL and HBase database to run on Hadoop (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M				L							

CO2	M				L							
CO3	S	M	S	M	S							
CO4	S	M	S	M	S							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
	Theory
Remember	20
Understand	30
Apply	50
Analyse	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List the similarities and differences between NoSQL vs SQL vs Hadoop. (Remember)
2. Explain the key differences in Replication and Sharding? (Understand)

Course Outcome 2 (CO2):

1. Illustrate the concept of bag in Pig? (Understand)
2. Explain about the different complex data types in Pig. (Understand)

Course Outcome 3 (CO3):

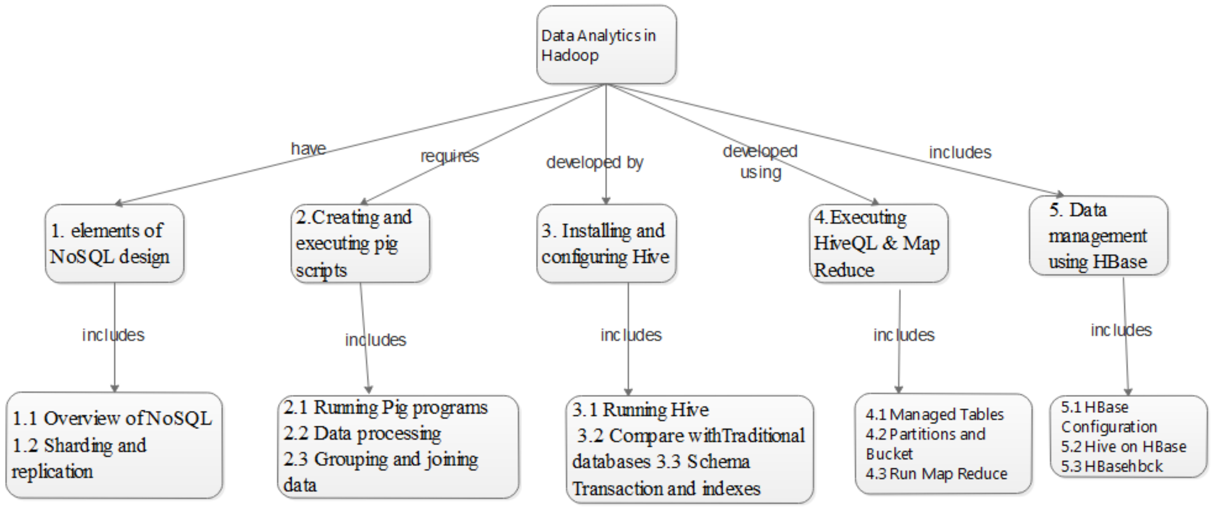
1. Identify whether we can change the data type of a column in a hive table? (Apply)
2. Construct a query to insert a new column (new_col INT) into a hive table (htab) at a position before an existing column (x_col). (Apply)

Course Outcome 4 (CO4):

1. Develop an application by connecting to Hbase?. (Apply)

2. Construct suitable commands to add a new column family “(newcolfamily”) to a table (“tablename”) which has a existing column family(“oldcolfamily”) (Apply)

Concept Map



Syllabus

NoSQL -Why NoSQL - Relational Databases- Persistent Data - Key Value and Document Data models - Column Family Stores - Graph databases - Schemaless Databases - Sharding - MAster slave replication - Peer-to-peer replication - Combining sharding and replication -CAP theorem

Pig - Execution Types - Running Pig Programs - Grunt - Pig Latin Editors - Pig Latin - Structure - Statements -Expressions - Types - Schemas - Functions - Macros - User-Defined Functions - Data Processing Operators - Loading and Storing Data - Filtering Data - Grouping and Joining Data - Sorting Data - Combining and Splitting Data - Pig in Practice - Parameter Substitution

Hive : Running Hive - Configuring Hive - Hive Services - The Metastore - Comparison with Traditional Databases - Schema on Read Versus Schema on Write - Updates, Transactions, and Indexes

HiveQL - Data Types - Operators and Functions - Tables - Managed Tables and External Tables - Partitions and Buckets - Storage Formats - Importing Data - Altering Tables - Dropping Tables - Querying Data - Sorting and Aggregating - MapReduce Scripts - Joins - Subqueries - Views - User-Defined Functions

HBase - Basic Hadoop/ZooKeeper/HBase configurations - High Availability (HA) masters - Data migration - Hbase Shell - HFile - Hive on HBase - HBasehbck - distcp - CopyTable - exporting - restoring and backing up - Basic performance tuning

References

1. Pramod J. Sadalage, Martin Fowler, NoSQL Distilled, Addison 2013

2. Yifeng Jiang, HBase Administration Cook Book, PACKT, 2012

3. Nick Dimiduk, AmandeepKhurana, HBase in Action, Manning, 2013

4. http://cassandra.apache.org/

5. https://docs.mongodb.com/

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**NoSQL**	
1.1	Why NoSQL – Open source databases-Operational Usecases	1
1.2	How Opensource NoSQL database is used in data analytics	1
2	**Pig**	
2.1	Execution Types - Running Pig Programs - Grunt - Pig Latin Editors - Pig Latin - Structure - Statements	1
2.2	Expressions - Types - Schemas - Functions - Macros - User-Defined Functions - Data Processing Operators - Loading and Storing Data - Filtering Data -	1
2.3	Grouping and Joining Data - Sorting Data - Combining and Splitting Data - Pig in Practice - Parameter Substitution	1
3	**Hive**	
3.1	Running Hive - Configuring Hive	1
3.2	Hive Services - The Metastore - Comparison with Traditional Databases	1
3.3	Schema on Read Versus Schema on Write - Updates, Transactions, and Indexes	1
4	**HiveQL**	
4.1	Data Types - Operators and Functions - Tables - Managed Tables and External Tables -	1
4.2	Partitions and Buckets - Storage Formats - Importing Data - Altering Tables - Dropping Tables - Querying Data	1
4.3	Sorting and Aggregating - MapReduce Scripts - Joins - Subqueries - Views - User-Defined Functions	1
5	**HBase**	
5.1	Basic Hadoop/ZooKeeper/HBase configurations	1
5.2	- High Availability (HA) masters - Data migration - Hbase Shell - HFile - Hive on HBase	1
5.3	HBasehbck - distcp - CopyTable - exporting - restoring and backing up - Basic performance tuning	1
	Total No of Hours	14

Course Designers:

1. K.Sundarakantham kskcse@tce.edu
2. Dr.B Muthukumaran dr.muthu@htcinc.com

		Category	L	T	P	Credit
14CS1J0	**CURRENT PRACTICES IN SOFTWARE ENGINEERING**	PE	1	0	0	1

Preamble

This subject is to promote the understanding of current practices in software engineering across phases from conceptualizing to operations and business process service delivery. The course is designed as an extension of the Software Engineering course and includes a project work in teams.

Prerequisite

Knowledge of Software Engineering

Course Outcomes

On the successful completion of the course, students will be able to

Explain the changes to Technology and Industry landscape (CO1)	Remember
Explain Impact of New Technologies / Industry Transformation on various phases of Software Lifecycle - From Requirements to Business Process Service Delivery (CO2)	Understand
Explain changes to software engineering practices to address changes to Technology and Industry landscape (CO3)	Understand
Plan, Build, Deliver, Maintain and Deliver Business Process Service (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L	L										
CO2	M	M				L	L	L			M	
CO3	M	M	M			L	L	L			L	M
CO4	S	S	S	L	S	M	M	M	S	S	S	S

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	20
Understand	20
Apply	60
Analyse	-
Evaluate	-
Create	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the major changes to the Technology landscape	Remember
2. What are the major changes to the Industry landscape	Remember
3. State about Hype-cycle	Remember

Course Outcome 2 (CO2):

1. What is the impact of changes to Technology and Industry landscape on Requirements, Design, Architecture	Remember
2. What is the impact of changes to Technology and Industry landscape on Build, Verification, Validation	Remember
3. Explain the impact of changes to Technology and Industry landscape on Maintenance and Business Process Service delivery	Understand

Course Outcome 3 (CO3):

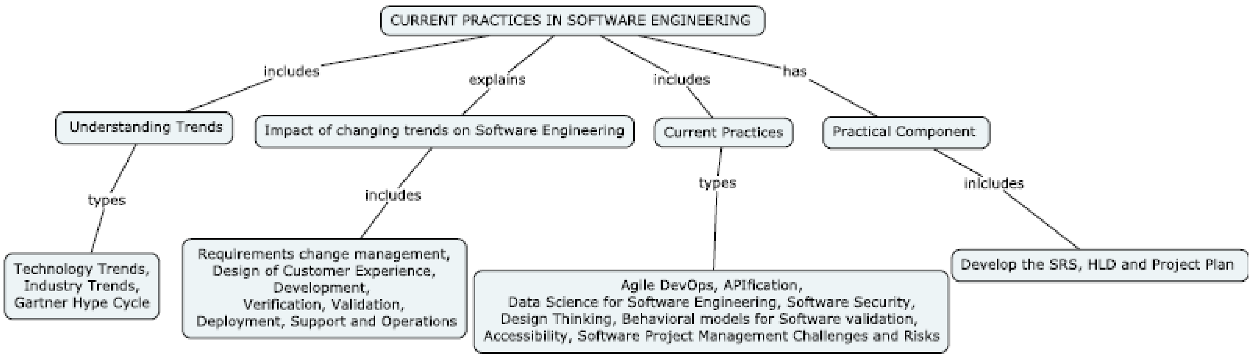
1. What are the current practices in Requirements, Architecture/Design	Remember
2. What are the current practices in Build, Verification, Validation	Remember
3. Explain the current practices in Maintenance and Business Process Service delivery	Understand

Course Outcome 4 (CO4):

A large retailer plans to adopt extensive use of IoT, Drones, knowledge of customers and AI; Help them build an inventory management system with the following requirements (a) Placing and Tracking orders; (b) Deliver the orders and Bill the customer; (c) Keep track of the stocks in the stores

1. Develop a modified architecture for the above system.	Apply
2. Develop a plan and design for the above system.	Apply

Concept Map



Syllabus

Understanding Trends:
Technology Trends: Transparently Immersive Experiences, Digital Platforms
Industry Trends: In Banking and Financial Services, Retail and Manufacturing
Gartner Hype Cycle: Hype Cycle Phases, Benefit Ratings, Maturity Levels

Impact of changing trends on Software Engineering:
Requirements change management, Design of Customer Experience, Development, Verification, Validation, Deployment, Support and Operations

Current Practices:
Agile DevOps, APIfication, Data Science for Software Engineering, Software Security, Design Thinking, Behavioral models for Software validation, Accessibility, Software Project Management Challenges and Risks

Practical Component
Develop the SRS, HLD and Project Plan considering that the application will extensively use of IoT, Drones, knowledge of customers and AI

Reference Books

1. Clean Architecture: A Craftsman's Guide to Software Structure and Designby Robert C Martin; Publisher: Prentice Hall; Release Date: September 2017; ISBN: 9780134494272
2. Designing Across Sensesby John Alderman, Christine W. Park; Publisher: O'Reilly Media, Inc.; Release Date: September 2017; ISBN: 9781491954249
3. Driving Digital by Isaac SacolickPublisher: AMACOMRelease Date: August 2017ISBN: 9780814438619
4. Clean Codeby Robert C. MartinPrentice Hall, 2008
5. Machine Learning Algorithmsby Giuseppe Bonaccorso; Publisher: Packt Publishing;Release Date: July 2017; ISBN: 9781785889622
6. Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Timeby Mark Raskino , Jackie FennPublisher: Harvard Business Review PressPublished: October 2008
7. Gartner Report “Hype Cycle for Emerging Technologies, 2017”Published: 21 July 2017 ID: G00314560Analyst(s): Mike J. Walker

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Understanding Trends(3)**	
1.1	Technology Trends: Transparently Immersive Experiences, Digital Platforms	1
1.2	Industry Trends: In Banking and Financial Services, Retail and Manufacturing	1
1.3	Gartner Hype Cycle: Hype Cycle Phases, Benefit Ratings, Maturity Levels	1
2	**Impact of changing trends on Software Engineering(3)**	
2.1	Requirements change management, Design of Customer Experience	1
2.2	Development, Verification, Validation	1
2.3	Deployment, Support and Operations	1
3	**Current Practices(4)**	
3.1	Agile DevOps, APIfication	1
3.2	Data Science for Software Engineering, Software Security	1
3.3	Design Thinking, Behavioural models for Software validation	1
3.4	Accessibility, Software Project Management Challenges and Risks	1
4	**Practical Component (4)**	
4.1	Develop the SRS, HLD and Project Plan considering that the application will extensively use of IoT, Drones, knowledge of customers and AI	4
	Total	**14**

Course Designers:

1. Mrs. Lakshmi V Murali Lakshmi.Murali@tcs.com
2. Mrs. A. Malini amcse@tce.edu

14CS1K0	AGILE SOFTWARE DEVELOPMENT & SAFE

Category	L	T	P	Credit
PE	1	0	0	1

Preamble

The course is designed to provide a solid foundation of Agile Scrum based software development. It highlights the key roles and ceremonies involved in Agile Scrum. Commercially successful large software projects involve a precise coordination of distributed Agile teams, hence the course throws light on SAFe (Scaled Agile Framework) as well.

Prerequisite

14CS571 - SOFTWARE ENGINEERING: THEORY AND PRACTICE

Course Outcomes

On the successful completion of the course students will be able to

Explain the various Scrum roles and Scrum ceremonies involved in agile based software development. (CO1)	Understand
Create Epics and detailed Use cases from the software requirements (CO2)	Apply
Prepare the estimates for the use cases, prioritize them in sprints, Monitor team velocity (CO3)	Apply
Illustrate principles in SAFe that enable distributed agile teams to work on a common Agile Release Train. (CO4)	Analyze
Create estimates and manage the burn down charts, measure and track Velocity to ensure an Agile Scrum project remains on schedule. (CO5)	Analyze

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M	M	L	L	M	L			M			L
CO2	M	M		L	M	M		M	M	M	M	L
CO3	M	M	M		M	M			M	M	M	M
CO4		M	M		M	M		M	M	M	M	M
CO5	S	M	M		M		M		M		M	M

S- Strong; M-Medium; L-Low

Assessment Pattern: Cognitive Domain

Cognitive Levels	Terminal Examination
Remember	0
Understand	30
Apply	40
Analyse	30

Evaluate	0
Create	0

Course Level Assessment Questions

Course Outcome 1(CO1):

1. Explain the main drivers for the evolution of Scrum based Software development and the principles behind Scrum
2. Highlight Differences between Agile Scrum and traditional models
3. A start-up is involved in the building of gaming apps. The competitors are rapidly gaining market in this space and the preferences of the players (kids to aged people) using the games is changing very fast. Justify why Agile Scrum will suit this scenario.

Course Outcome 2(CO2):

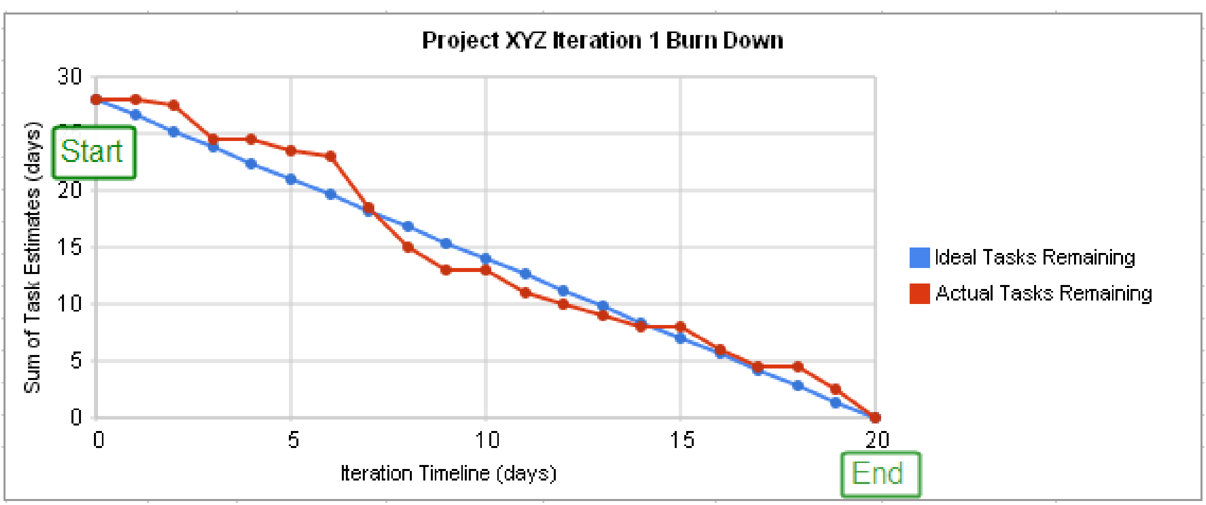
1. Describe the various roles in Agile Scrum process and explain why scrum master's role is more than that of a traditional manager's role
2. Highlight Differences between sprint backlog and product backlog
3. Explain the elements in "Definition of Done"
4. During the 3rd Sprint of a software development, the customer has changed a major requirement. Explain the process involved in handing the change.

Course Outcome 3(CO3):

1. Highlight the importance of having Daily Scrum Meetings and explain the purpose it solves.
2. Explain the need for refactoring the code.
3. In the PDCA cycle, where does Sprint Retrospective meeting fits.

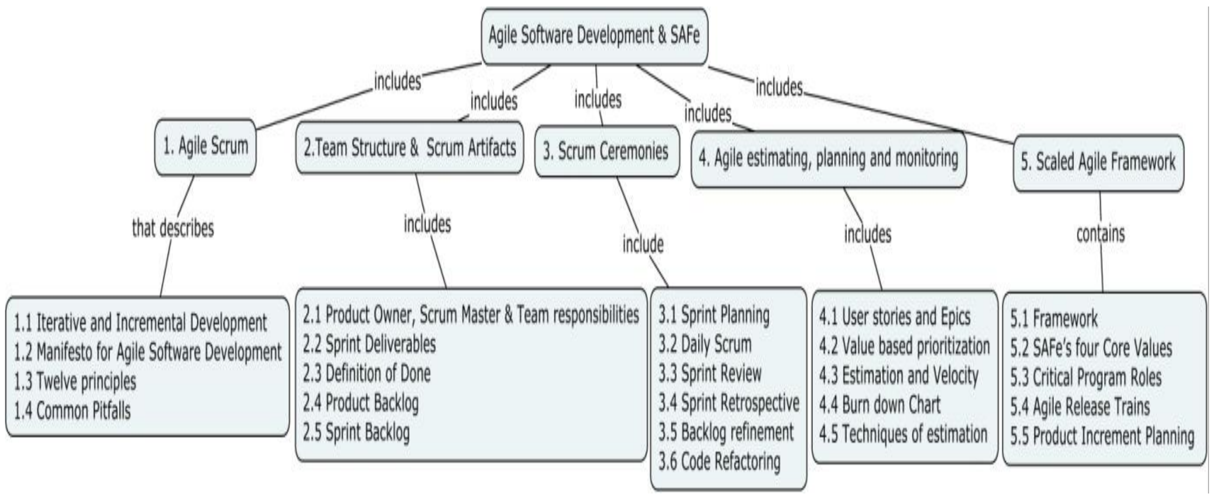
Course Outcome 4 (CO4):

1. Your organization develops Hospital management software for small clinics. Create the Epics and Use cases.
2. Is it right to measure the velocity of a team at the end of the first sprint? Justify your answer with examples.
3. Take a look at the below burn down chart and comment at the project progress at various time periods.



Course Outcome 5 (CO5):
1. How does SAFe differ from Agile Scrum
2. How is Program Increment (PI) related to ART?
3. What do you mean by Scrum of Scrums

Concept Map



Syllabus

Agile Scrum --- A Short History of Agile - Iterative and Incremental Development - Manifesto for Agile Software Development - 12 principles - Common Pitfalls

Team Structure and Artefacts --- Product Owner - Scrum Master & Team responsibilities - Sprint Deliverables - Definition of Done - Product Backlog - Sprint Backlog

Scrum Ceremonies --- Sprint Planning - Daily Scrum - Sprint Review & Sprint Retrospective - Backlog refinement - Code Refactoring

Agile estimating, planning and monitoring --- User stories and Epics - Value based prioritization -Estimation and Velocity - Burn down Chart - Techniques of estimation - Planning poker and Affinity estimation

Scaled Agile Framework (SAFe) --- Introducing the Scaled Agile Framework (SAFe) - SAFe's 4 Core Values - Critical Program Roles - Agile Release Trains (ARTs) - Common principles - Program Increment Planning

References

1. Mary and Tom Poppendieck, Implementing Lean Software Development, Addison Wesley
2. Mike Cohn,Succeeding with Agile, Software Development with Scrum, Addison Wesley
3. https://scaledagile.com

Course Contents and Lecture Schedule

Module Number	Topic	No. of Lectures
1	**Agile Scrum**	
1.1	A Short History of Agile - Iterative and Incremental Development	1
1.2	Manifesto for Agile Software Development	1
1.3	12 principles - Common Pitfalls	1
2	**Team Structure & Scrum Artefacts**	
2.1	Product Owner- Scrum Master & Team responsibilities	1
2.2	Sprint Deliverables - Definition of Done	1
2.3	Product Backlog - Sprint Backlog	1
3	**Scrum Ceremonies**	
3.1	Sprint Planning- Daily Scrum- Sprint Review & Sprint Retrospective.	1
3.2	Backlog refinement- Code Refactoring	1
4	**Agile estimating, planning and monitoring**	
4.1	User stories and Epics - Value based prioritization	1
4.2	Estimation and Velocity - Burn down Chart	1
4.3	Techniques of estimation: Planning poker and Affinity estimation	1
5	**Scaled Agile Framework (SAFe)**	
5.1	Introducing the Scaled Agile Framework (SAFe) - SAFe's four Core Values	1
5.2	Critical Program Roles	1
5.3	Agile Release Trains (ARTs) - common principles- Program Increment Planning	1
	Total Hours	**14**

Course Designers:

1. G.S.R. Emil Selvan emil@tce.edu
2. M.P.Ramkumar ramkumar@tce.edu
3. M. Madhan madhan.mahalingam@honeywell.com

14CS1L0	HETEROGENEOUS COMPUTING	Category	L	T	P	Credit
		PE	1	0	0	1

Preamble

This course will walk the students through the power of the emerging high performance computing systems and let them learn three major parallel programming approaches such as OpenMP, Message Passing Interface and CUDA programming.

Prerequisite

C Programming

Course Outcomes

On the successful completion of the course, students will be able to

Develop shared memory parallel program using OpenMP	(CO1)	Apply
Develop distributed memory parallel program using MPI	(CO2)	Apply
Develop data parallel program using CUDA	(CO3)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO3.	S											
CO4.		M		S								
CO3			S									

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Terminal Examination
Remember	10
Understand	30
Apply	60
Analyse	0
Create	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Which type of applications is best suited for shared memory model?
2. What are the different metrics for assessing a parallel program?

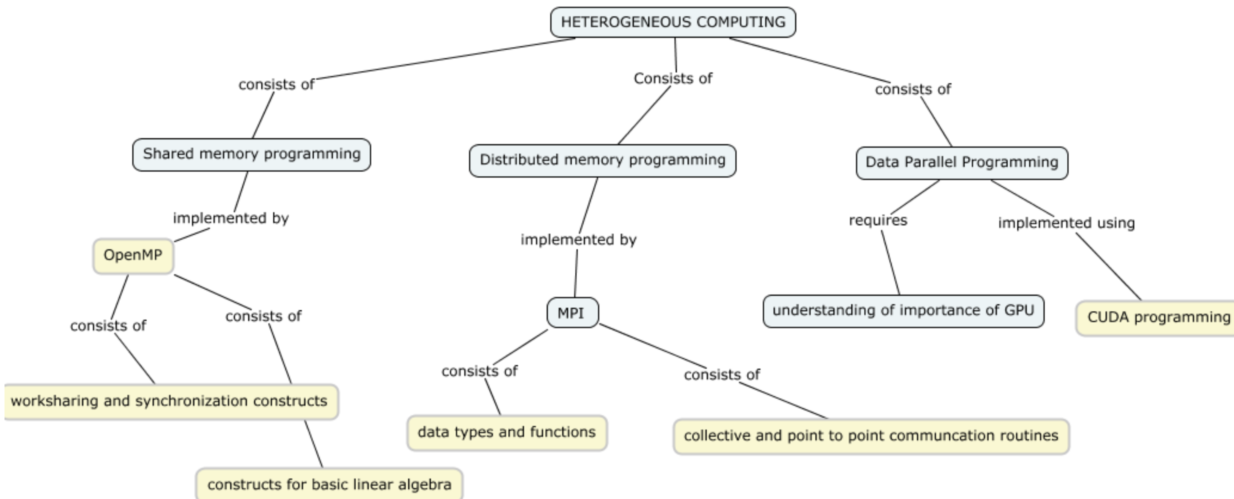
Course Outcome 2 (CO2):

1. Write an example program to illustrate the usage of MPI_Barrier.
2. List the differences between collective and point to point communication

Course Outcome 3 (CO3):

1. How many different kinds of memory are in GPU
2. Which algorithm performs better on the GPU? CPU bound or data bound?

Concept Map



Syllabus

Introduction to parallel programming paradigms

Introduction to the shared memory programming- Distributed memory parallel programming- Data parallel programming model- Performance metrics.

Shared Memory Programming

Introduction to OpenMP- Constructs –Parallel, work sharing and Synchronization constructs- Basic linear algebra operations in OpenMP

Distributed memory Programming

MPI data types and tags- compiling MPI programs- MPI functions- Collective communication- Point to Point communication- MPI numerical Integration case studies

Data Parallel Programming

Importance of data parallelism for GPUs- introduction to CUDA-Memory and variable types- control flow- synchronizations- wrap shuffles-reduction operations

References:

- https://computing.llnl.gov/tutorials/openMP/
- https://www.dartmouth.edu/~rc/classes/intro_mpi/
- https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

Course Contents and Lecture Schedule

S.No	Topic	No. of Lectures
1	**Introduction to Parallel Programming**	
1.1	What is Parallelization- Goals	

1.2	Performance metrics for parallelization	1
1.3	Different parallel programming models	
1.3.1	Shared memory model	2
1.3.2	Distributed memory model	
1.3.3	Data parallel model	
2	**Shared Memory programming**	
2.1	Introduction to OpenMP and its constructs	1
2.1.1	Basic constructs of OpenMP	
2.1.2	Synchronization constructs	1
2.1.3	Environment variable constructs	
2.1.4	Basic linear algebra operations in OpenMP	1
3	**Distributed memory programming**	
3.1	Introduction to MPI	1
3.1.1	MPI data types and tags- compiling MPI programs	
3.1.2	MPI functions- Collective communication- Point to Point communication	2
3.1.3	MPI numerical Integration case study	2
4	**Data Parallel Programming**	
4.1	Importance of data parallelism for GPUs	1
4.2	Introduction to CUDA-Memory and variable types-control flow-synchronizations- wrap shuffles-reduction operations	3
	Total No. of. hours	**15**

Course Designers:

1. Mr.Ashok Chaudhry, Fujitsu Ashok.Chaudhary@ts.fujitsu.com
2. Dr.P.Chitra,TCE pccse@tce.edu

14CS2A0	MULTICORE INTERCONNECTS: THEORY AND PRACTICE	Category	L	T	P	Credit
		PE	2	0	0	2

Preamble

Storage and Communication play a crucial role in the design and performance of multi-core Systems-On-Chips (SoCs). Network On Chip (NoC) has been proposed as a potential solution to simplify and optimize SoC design.

This course explores (i) cache memory concepts and its optimization techniques, (ii) design concepts in memory controllers and mass storage structures, and (iii) principles and practices of Networks-on-chip (NoC).

The course also focuses on building a strong theoretical foundation for NoC, besides giving an exposure to the state of the art architectural tools for simulating cache and NoC for multicore systems.

Prerequisite

- 14CS540 – Computer Architecture

Course Outcomes

On the successful completion of the course, students will be able to

CO1:	Apply appropriate optimizations techniques for improving the performance of caches	Apply
CO2:	Summarize the basic working and organization of Dynamic RAM and memory controllers	Understand
CO3:	Examine the performance characteristics of various routing algorithms used in Network On Chip architectures.	Analyze
CO4:	Analyze the different adaptive techniques used for the avoidance of congestion in multicore chips	Analyze

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	S	L	M							
CO2	M	M	L									
CO3	S	S	S	M	M				M			
CO4	S	S	S	M	M							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	CAT - I	Terminal Examination
Remember	20	20
Understand	30	20
Apply	30	30
Analyse	20	30

Evaluate	-	-
Create	-	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Assume a memory model where you have a cache size of 4 blocks, a block size of 16 words/block, and addresses from 0 through 511 in main memory. a) What addresses would be in the same block as address 201 in main memory? Assume that the cache is empty before the start of the sequence in each case. Assume that the 2way associative cache uses a LRU (last recently used) eviction strategy. Consider the following sequence of memory accesses:
   18, 70, 4, 16, 65, 10, 84, 470, 12, 475, 90, 70
   Identify if each memory access in the sequence would be a hit or a miss for a direct mapped cache, 2-way set associative cache.
2. Consider two cache architectures. One has separate I and D cache of size 16KB each and the other one is unified dual ported of size 32KB. The split caches has instruction miss rate 0.5% and data miss rate 5%. The unified cache has aggregate miss rate 2%. Hit time is 1 cycle. Miss penalty is 50 cycle. 30% of instructions are load/store. Which one is better and what is the improvement in CPI ? Assume CPI of 1 without cache misses.
3. A cache has access time (hit latency)=10 ns and miss rate is 5%. An optimization was made to reduce the miss rate to 3 % but the hit latency was increased to 15 ns. Under what condition this change will result in better performance (Lower avg. memory access time)?

Course Outcome 2 (CO2):

1. What is refreshing and refresh overhead in DRAM?
2. Explain open row buffer scheduling policy?

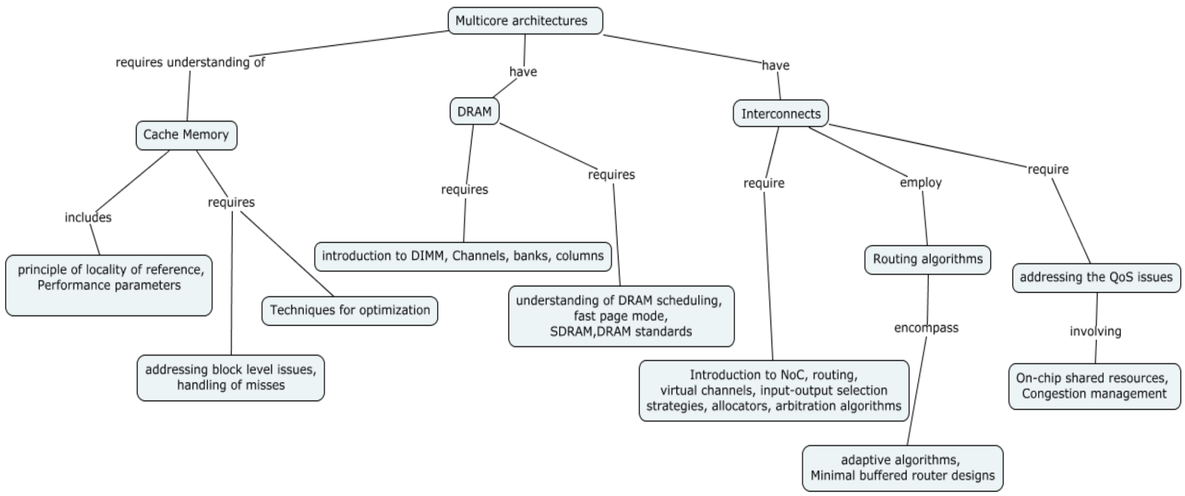
Course Outcome 3 (CO3):

1. Implement odd even routing in the Garnet simulator and find out percentage reduction in saturation latency with respect to XY routing in a 4x4 and 6x6 mesh NoC. Use (a) uniform traffic (b) tornado traffic
2. Find out the number of conflict misses when two high MPKI SPEC benchmark applications are ran on a Gem5 –ruby set up. The processor is dual core with 16KB direct mapped I cache, 16 KB 2 way, D-cache, 64 KB L2 cache (4 way associative, 8B block size).

Course Outcome 4 (CO4)

1. Find out the link utilization factor in and 8x8 mesh NoC that carries transpose traffic pattern and identify the hotspots. Vary the number of input buffers in the NoC router and find out the saturation point. Collect the statistics from Garnet simulator.
2. Assess the impact of different priority mechanism in deflection routers.
3. Explain why the silver flit scheme in MinBD router is counterproductive.
4. List out few congestion metrics used in NoC. Illustrate the merits and limitations of each of each of this by proper statistics collected from real time simulations from Gem5-Ruby-garnet tool.

Concept Map



Syllabus

Introduction: memory hierarchy, locality of reference, cache memory fundamentals, cache performance parameters. Block level issues -mapping, identification, replacement, write strategy, types of misses-compulsory, capacity, conflict misses. Basic cache optimizations by adjusting cache size, block size, associativity. **Introduction to DRAM organization**- DIMMs, channels, ranks, banks, rows, columns. x2, x4, x8, xn devices and bandwidth/capacity calculations. DRAM controller -scheduling and refreshing policies. Fast page mode, SDRAM, DDR SDRAM, modern DRAM standards. DIMM based design. **Network on Chip (NoC):** Introduction to NoC, topology, routing, flow control, virtual channels, input buffered router micro-architecture. Input and output selection strategies, allocators and arbiter algorithms for crossbar switch. **Routing in NoC:** Various types of routing algorithm- Adaptive routing algorithm. Introduction to deflection routers- minimally buffered router designs. **QoS in NoC:** Need for QoS on-chip shared resources like LLC, NoC and MC. Congestion in NoC and Congestion awareness strategy.

Hands-on Topics:

1. Installation / Understanding gem5-ruby-garnet (3 hours)
2. Study of effect of block size and associativity on CPI (2 hours)
3. Study of cache replacement algorithms on CPI (1 hour)
4. Analysis of cache misses (1 hour)
5. Implementation of static and adaptive NoC routing algorithms (2 hours)
6. Packet latency and network path analysis in NoC (3 hours)
7. Cache miss aware routing techniques. (2 hours)

References

1. Hennessey and Patterson: “Computer Architecture A Quantitative Approach”, 5th Edition, Elsevier,2012
2. Bruce Jacob, Spencer W. Ng, David T. Wang, Memory System-Cache, DRAM and Disk Morgan Kaufman,2008.
3. William James Dally, Brian Towles, Principles and Practices of Interconnection Networks, Morgan Kaufman, 2004

4. http://www.gem5.org/Main_Page
5. https://www.spec.org/cpu2006/

Course Contents and Lecture Schedule

Module .No	Topics	No. of Lectures
1.	Introduction	
1.1	memory hierarchy, locality of reference, cache memory fundamentals, cache performance parameters	3
1.2	Block level issues -mapping, identification, replacement, write strategy, types of misses-compulsory, capacity, conflict misses	
1.3	Basic cache optimizations by adjusting cache size, block size, associativity.	
2.	**Introduction to DRAM organization**.	2
2.1	DIMMs, channels, ranks, banks, rows, columns. x2, x4, x8, xn devices and bandwidth/capacity calculations	
2.2	DRAM controller, scheduling and refreshing, Fast page mode, SDRAM, DDR SDRAM, modern DRAM standards. DIMM based design	
3	**Network on Chip (NoC)**	3
3.1	Introduction to NoC, topology, routing,	
3.2	Flow control, virtual channels, input buffered router micro-architecture.	
3.3	Input and output selection strategies, allocators and arbiter algorithms for crossbar switch.	
4.	**Routing in NoC**	3
4.1	Various types of routing algorithm- Adaptive routing algorithm	
4.2	Introduction to deflection routers- minimally buffered router designs	
5.	**QoS in NoC**	3
5.1	Need for QoS	
5.2	QoS on on-chip shared resources like LLC, NoC and Memory Controller	
5.3	Congestion in NoC	
5.4	Congestion awareness strategy	
6.	Total no. of hands -on hours	14
	Total Lecture Hours	**28**

Course Designers:

1. Dr. John Jose johnjose@iitg.ernet.in
2. Dr.P.Chitra pccse@tce.edu

14CS2B0	**DATA ANALYTICS USING R AND PYTHON**	Category	L	T	P	Credit
		PE	2	0	0	2

Preamble

This course explores (i) Basic understanding about Machine Learning (ii) Basic R and Python Skills to continue learning ML concepts and participate in labs (iii) Evaluate and recommend best fit models for financial and statistical data.

The course also focuses on building a proficiency in SQL, and basic structured data manipulation, software programming skills/aptitude to basic R programming.

Prerequisite

- 14CS310 – Probability and Statistics
- Matrices, Vectors, Linear Algebra
- Proficiency in SQL and basic structured data manipulation

Course Outcomes

On the successful completion of the course, students will be able to

Compare and contrast the types of learning methods, improving models, Evaluating the model performance (CO1)	Understand
Apply the appropriate machine learning concepts to possible use cases in their field of expertise (CO2)	Apply
Apply Machine learning techniques using R and Python for text and statistical Data (CO3)	Apply
Identify best fit models for financial data set using ML techniques (CO4)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO5.	S	M	L									
CO6.	S	M	M	L	M							
CO3.	S	S	S	M	S	M		L				L
CO4.	S	S	S	M	S	L	L		M	M	M	M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	CAT – 1	Terminal Examination
Remember	20	20
Understand	30	30
Apply	50	50

ANALYSE	-	-
EVALUATE	-	-
CREATE	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare R and Python programming languages for Predictive Modelling
2. In base graphics system, which function is used to add elements to a plot?
3. What is meant by K-nearest neighbour?

Course Outcome 2 (CO2):

1. Write code to sort a DataFrame in Python in descending order
2. Consider the Naive Bayes algorithm when trained on the following dataset:

X1	X2	X3	LABEL
1	1	1	+
1	0	1	+
1	1	0	+
1	0	0	-
0	0	0	-

3. What prediction does it make on the instance x1 = 1, x2 = 0, x3 = 0.

Course Outcome 3 (CO3):

1. How to create scatter plot matrices in R language?
2. How to merge two data frames in R programming language?
3. Write the R and Python programming code for an array of words so that the output is displayed in decreasing frequency order

Course Outcome 4 (CO4):

1. Apply the ID3 supervised learning algorithm to find the classification rule for genuine customer from the given table.

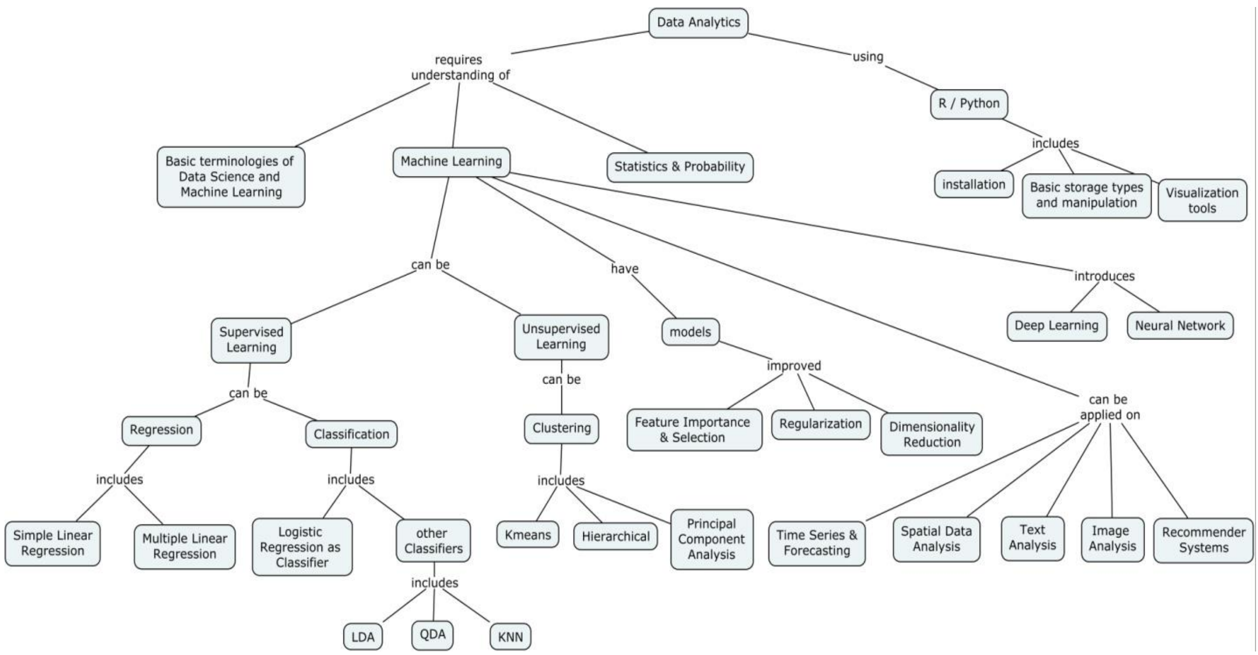
age	salary	Repayment
40	55	yes
36	51	yes
56	71	no
50	67	yes
47	57	no
34	67	yes
54	82	no

2. The given table the first column shows scores on the aptitude test and the second column

shows statistics grades of students performance. Apply regression analysis for the given.

Aptitude Test	statistics grades
70	55
96	85
90	95
60	67
70	87
80	87

Concept Map



Syllabus

Mathematical Foundations: Linear Algebra, Matrices, Vectors, Statistics and Probability**.** **Introduction to R Programming:** Installation, Basic storage types and manipulation, Basic Visualization tools. **Introduction to Python Programming:** Installation, Basic storage types and manipulation, Basic Visualization tools**. Basic Terminologies relating to Data Science and Machine Learning:** Key terminologies, Data Science for Process Model. **Introduction to Machine Learning:** Supervised Learning, Unsupervised Learning. **Supervised Learning (Regression):** Simple Linear Regression, Multiple Linear Regression. **Supervised Learning (Classification):** Classifiers, Logistic Regression as a classifier. **Unsupervised Learning**

(Association Technique and Clustering): K-means clustering models, Hierarchical Clustering models, Principal Component Analysis. **Improving ML Models & Glimpse of Advanced Topics:** Feature selection, Regularization, Dimensionality Reduction. **Advanced Topics in ML:** Scenarios for Applying ML, Deep Learning, Neural Networks.

Hands-on Topics:

- Demo / Tutorials on R Programming
- Demo / Tutorials on Python Programming

Reference Books

Mathematics for Machine Learning:

- http://courses.washington.edu/css490/2012.Winter/lecture_slides/02_math_essentials.pdf
- https://datascience.ibm.com/blog/the-mathematics-of-machine-learning/

R:

- https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
- https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

Python:

1. Introducing Python by Bill Lubanovic
2. Introduction to Machine Learning with Python:A guide for Data Scientists by Andreas C Muller
3. https://www.cs.uky.edu/~keen/115/Haltermanpythonbook.pdf

Course Contents and Lecture Schedule

Module No.	Topics	No. of Lectures
1.	**Statistics & Probability**	2
2.	**Introduction to R programming**	6
2.1	Installation & exploring the IDE	
2.2	Basic storage types and manipulation	
2.3	Basic Visualization tools	

3.	**Introduction to Python Programming**	6
3.1	Installation & exploring the IDE	
3.2	Basic storage types and manipulation	
3.3	Basic Visualization tools	
4.	**Basic Terminologies relating to Data Science and Machine Learning**	1
4.1	Key terminologies	
4.2	Data Science for Process Model	
5.	**Introduction to Machine Learning:**	1
5.1	Supervised Learning	
5.2	Unsupervised Learning	
6	**Supervised Learning (Regression)**	3
6.1	Simple Linear Regression	
6.2	Multiple Linear Regression	
7	**Supervised Learning (Classification)**	3
7.1	Classifiers	
7.2	Logistic Regression as a classifier	
8	**Unsupervised Learning (Association Technique and Clustering)**	2
8.1	K-means clustering models	
8.2	Hierarchical Clustering models	
8.3	Principal Component Analysis	
9	**Improving ML Models & Glimpse of Advanced Topics**	2
9.1	Feature selection	
9.2	Regularization	
9.3	Dimensionality Reduction	
10	**Advanced Topics in ML**	2
10.1	Scenarios for Applying ML	
10.2	Deep Learning	
10.3	Neural Networks	
Total Lecture Hours		**28**

Course Designers:

1. Mr.Vidhya Shankar Venkatesan vidhya.venkatesan17@iimb.ac.in
2. Mr.Sundara Raman Narayanan sundara.narayanan17@iimb.ac.in
3. Mr.Srinivasan Sridhar srinivasan.sridhar@iiml.org

4. Dr. C. Deisy cdcse@tce.edu
5. Dr.S.Padmavathi spmcse@tce.edu

		Category	L	T	P	Credit
14CSGA0	**WEB TECHNOLOGIES**	GE	3	0	0	3

Preamble

This course is offered in the Seventh semester for the students of Under Graduates. The students will learn how to represent the structure and designs using HTML and XHTML and other related web technologies. The students gains understanding of how the internet application works and develop web programming skills. The course will establish a professional, client-based attitude towards web design

Course Outcomes

On the successful completion of the course, students will be able to

Understand the working of internet with HTML and XHTML. (CO1)	Understand
Understand the working principle of internet applications. (CO2)	Understand
Construct HTML and XHTML documents. (CO3)	Apply
Develop client/ server side programming for web application Development.(CO4)	Apply
Develop web pages with database connectivity.(CO5)	Apply

Mapping with Programme Outcomes

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1.	M											
CO2.	M											
CO3.		M	M						S	S	S	M
CO4.		M	M						S	S	S	M
CO5.		M	M						S	S	S	M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**			**Terminal Examination**
	1	**2**	**3**	
Remember	20	20	20	20
Understand	60	20	20	20
Apply	20	40	60	60
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define IP Addressing
2. Define Computer Network.
3. List the types of Domain Name Space.

Course Outcome 2 (CO2):

1. Sketch the Web Browser and Server Communication.
2. Write the Structure of HTML.
3. List out the version of HTML.

Course Outcome 3 (CO3):

1. Draw the XHTML for Book Publisher along with List of Books.
2. Explain in detail about Servlet Database Connectivity with an example of Student database.
3. Write a servlet program to display the message by using Cookies.

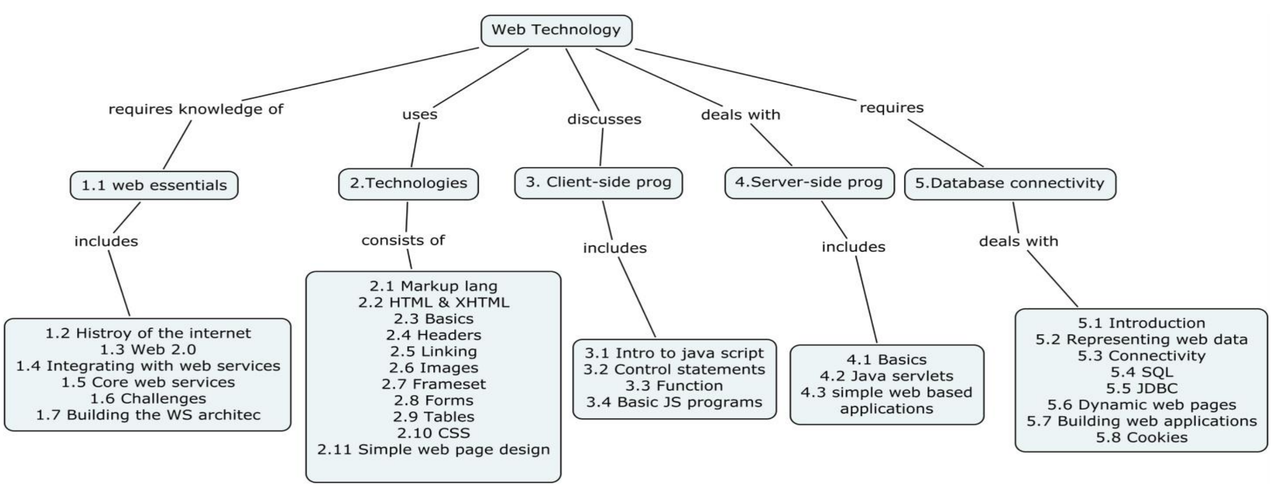
Course Outcome 4 (CO4):

1. Write a Java Script program to print the numbers from 0 to 50. b. Write a Java Script program to create table.
2. Explain in detail about Array with an example of Positive and negative numbers.
3. Write a Java Script program to create user registration form.

Course Outcome 5 (CO5):

1. Construct the book's HTML document using cascading style sheets.
2. Draw a picture describing the relationship between client/server objects used by SQL
3. Write a Java Script program to create student registration form using database connectivity.

Concept Map



Syllabus

Introduction to Web - Web essentials – Basics of Internet - History of the Internet and World Wide Web - Web 2.0- Technology overview - integrating with web services - Motivation and Characteristics – uses - Basic operational model of web services - core web services – Known challenges in Web Services - Building the Web Services Architecture. **Technologies** – Markup languages – HTML & XHTML- Basics – Headers – Linking –Images – Frames –Frameset – Forms – Tables – CSS – simple web page designs. **Client-side Programming –** Introduction to Java script – Control statements – Function – Basic Java script programs. **Server-side programming** - Server side programming basics – java servlets – simple web based applications- Session – Session tracking. **Database Connectivity -** Introduction to Database-Representing Web data – data base connectivity –SQL/MS-Access- Dynamic Web pages – Building Web applications – cookies.

Text Books

1. Deitel and Deitel, "Internet and World Wide Web How to program", Prentice Hall of India, Fourth Edition, 2004.
2. Gustavo Alonso, Fabio Casati, Harumi Kuno and Vijay Machiraju, "Web services" Springer International Edition, First edition, 2009.

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction to Web**	
1.1	Web essentials and Basics of Internet	1
1.2	History of the Internet and World Wide Web	1
1.3	Web 2.0- Technology overviews	1
1.4	Integrating with web services - Motivation and Characteristics - Basic operational model of web services - uses	2
1.5	Core web services	1
1.6	Challenges in Web Services	1
1.7	Building the Web Services Architecture	1
2	**Technologies**	
2.1	Markup languages	1
2.2	HTML & XHTML	1
2.3	Basics	2
2.4	Headers	1
2.5	Linking	1
2.6	Images	1

2.7	Frames - Framesets	1
2.8	Forms	1
2.9	Tables	1
2.10	CSS –Cascading style sheets	2
2.11	Simple web page designs	1
3	**Client-side Programming**	
3.1	Introduction to Java script	1
3.2	Control statements	2
3.3	Function	2
3.4	Basic Java script programs	1
4	**Server-side programming**	
4.1	Server side programming basics	1
4.2	Java Servlets	2
4.3	Simple web based applications - Session	2
Database Connectivity		
5.1	Introduction to Database, Representing of Web data	1
5.2	Data base connectivity	1
5.3	SQL/MS-Acess	1
5.4	Dynamic web pages, Building Web applications, Cookies	1
	Total	36

Course Designer

1. Mr. T.Manikandan tmcse@tce.edu

14CSGB0	**ESSENTIALS OF MOBILE APPLICATION DEVELOPMENT**	Category	L	T	P	Credit
		GE	3	0	0	3

Preamble

Mobile Technologies for Smart Phones and Tablets like iPhone, iPad, Android etc., are the next big thing on Information Technology (IT) and as well as Telecom horizons. This course shall provide specialized knowledge on computing with focus on mobile application technology. Students will be trained in understanding the concepts of emerging technologies in development of mobile applications. He / She will learn the fundamental principles to design and develop a mobile application using android platform.

Prerequisite

- Basics of Objected Oriented Programming
- Basics of Computer Networks

Course Outcomes

On the successful completion of the course, students will be able to

Describe the major mobile device platforms and their capabilities.(CO1)	Understand
Apply the principles of web service architecture to mobile application development.(CO2)	Apply
Explain the mobile platforms iOS and Android SDK.(CO3)	Understand
Illustrate the mobile application architecture of Android.(CO4)	Understand
Apply techniques of Android to leverage the design of an application. (CO5)	Apply

Mapping with Programme Outcomes

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1.	M				M							
CO2.	M	M	M		M				M	M	L	
CO3.	M				M							
CO4.	M											
CO5.	M	L	L		L				L	M	L	

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**			**Terminal Examination**
	1	**2**	**3**	
Remember	40	30	20	20

Understand	20	30	40	40
Apply	40	40	40	40
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Draw the Android OS architecture.
2. Define mobile computing.
3. State the purpose of Layouts.
4. Define REST.
5. List the application components of Android.

Course Outcome 2 (CO2):

1. Design a REST service to receive data from the notes mobile application and save it on a server database.
2. Revise the notes mobile application to communicate with the server over REST protocol and save the notes on the server.
3. Develop a simple iOS notes application to take notes based on the text entered by the user and store the notes in a database.
4. Practice XCode IDE on Apple Mac Desktop and Eclipse IDE on Windows Desktop

Course Outcome 3 (CO3):

1. Explain Apple (iOS) and Android Ecosystem.
2. Describe the various technology options available for development like native development, cross platform development platforms, rich web.
3. Distinguish the different Integrated Development Environments (IDE)s available for mobile application development.
4. Compare the different ways to design a mobile application.

Course Outcome 4 (CO4):

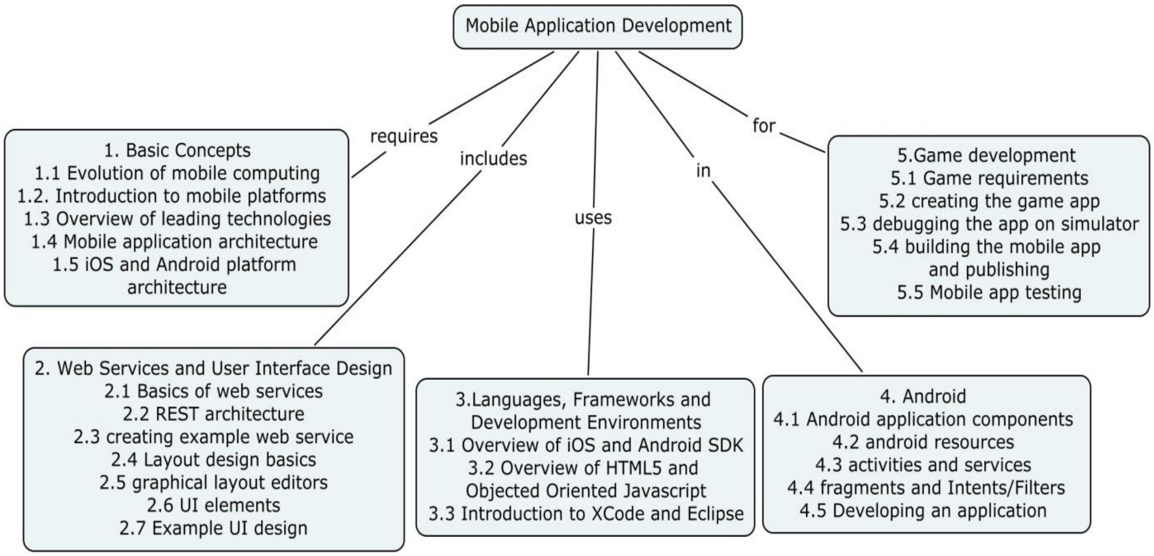
1. Demonstrate the different ways to design a mobile application using Android SDK and HTML5.
2. Explain the online guides from Apple and Google on iOS and Android SDK.
3. Explain the device capabilities of iOS and Android based smart phones and Tablets.
4. Illustrate the steps to launch an mobile application on Android platform.

Course Outcome 5 (CO5)

1. Facilitate the smart phone's camera to take a photo and attach with the note and get the user's current location using the GPS available on the smart phone.
2. Design a mobile web application with suitable UI Components.

3. Assume specific game requirements and design a Game application using Android framework.

Concept Map



Syllabus

Basic Concepts: Evolution of mobile computing, Introduction to mobile platforms, Overview of leading technologies available for mobile development, mobile application architecture – iOS and Android. **Web Services and User Interface Design:** Basics of web services, REST architecture, creating example web service, Layout design basics, graphical layout editors, UI elements, Example UI design. **Languages, Frameworks and Development Environments:** Overview of iOS and Android SDK, Overview of HTML5 and Objected Oriented Javascript, Introduction to XCode and Eclipse development environments and device simulators. **Android basics:** Android application components, android resources, activities and services, fragments, Intents/Filters, Developing an application in Android. **Game development using Android:** Game requirements, creating the game app, debugging the app on simulator, building the mobile app and publishing to the market, testing the mobile app.

Reference Books

1. Jeff McWherter, Scott Gowell, “Professional Mobile Application Development”, John Wiley & Sons, Inc., 2012.
2. Mike Wolfson, “Android Developer Tools Essentials”, O’Reilly, 2013.
3. Derek James, “Android Game Programming For Dummies”,John Wiley & Sons, Inc., 2013.
4. Android Developer’s Guide - http://developer.android.com/guide/index.html

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**BASIC CONCEPTS (5)**	
1.1	Evolution of mobile computing	1
1.2	Introduction to mobile platforms	1
1.3	Overview of leading technologies available for mobile development	1
1.4	Mobile application architecture- iOS	1
1.5	Mobile application architecture- Android	1
2.	**WEB SERVICES AND USER INTERFACE DESIGN(10)**	
2.1	Basics of web services	1
2.2	REST architecture	2
2.3	Creating example web service	1
2.4	Layout design basics	2
2.5	Graphical layout editors	1
2.6	UI elements	2
2.7	Example UI design	1
3.	**LANGUAGES, FRAMEWORKS AND DEVELOPMENT ENVIRONMENTS(6)**	
3.1	Overview of iOS SDK	1
3.2	Overview of Android SDK	1
3.3	Overview of HTML5	1
3.4	Overview of Objected Oriented Javascript	1
3.5	Introduction to XCode and Eclipse development environments and device simulators	2
4.	**ANDROID BASICS(8)**	
4.1	Android application components	1

Module No.	Topic	No. of Lectures
4.2	Android resources	1
4.3	Activities and services	2
4.4	Fragments	1
4.5	Intents/Filters	1
4.6	Developing an application in Android	2
5.	**GAME DEVELOPMENT USING ANDROID (7)**	
5.1	Game requirements	1
5.2	Creating the game app	2
5.3	Debugging the app on simulator	1
5.4	Building the mobile app and publishing to the market	2
5.5	Testing the mobile app	1
	Total	**36**

Course Designer:

1. G.Madhu Priya gmadhupriya@tce.edu

		Category	L	T	P	Credit
14CSGC0	**ANIMATION : THEORY AND PRACTICE**	GE	3	0	0	3

Preamble

The goal of this subject is to motivate the students to acquire knowledge and skills in 2D and 3D animation. The topics like 2D, 3D objects and visualization help the students to implement their own ideas with more creativity.

Prerequisite

Problem Solving using Computers

Course Outcomes

On the successful completion of the course, students will be able to

Produce an object after applying the required 2D/ 3D transformation techniques. (CO1)	Apply
Identify the visible surface and invisible surfaces of 3D objects by applying suitable surface detection algorithms. (CO2)	Apply
Develop 2D / 3D animation for a given scenario by applying the principles of animation. (CO3)	Apply
Identify suitable visualization techniques for the given problem. (CO4)	Apply
Explain the need for projections, modeling, rendering and texturing. (CO5)	Understand

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S	S	L		L			L	M	L		
CO2.	S	M	L		L							
CO3.	S	S	M		S			L	M	L		M
CO4.	M	M	L									
CO5.	M	L			L							

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	30	30	20	20
Apply	40	50	60	60
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Reflect a diamond shaped polygon whose vertices are A (-1,0), B(0,-2), C(1,0) and D (0,2) about
   i) the Horizontal Line y= 3
   ii) the vertical line x=4
   iii) the line y=x+3
2. Illustrate the effect of following shearing transformations on the square A (0, 0), B (4, 0), C (4, 4) and D (0, 4)
   i) X direction shearing when $sh_x = -6$
   ii) Y direction shearing with respect to $x_{ref} = 3$ and $sh_y = 3$
   iii) Shearing in both directions when $sh_x = 5$ and $sh_y = 5$
3. Distinguish coordinate from geometric transformation.
4. Determine the new co-ordinate position on the following points A(4,3,8) , B(-7,5,8) and C(6, 8,-4) after scaling w.r.to pivot point (3,4,3) and the scaling factor is given by (8,8,5).
5. Perform 3D rotation on a unit cube about y axis w.r.to pivot point (4, 0,-7) and θ =60 degree.

Course Outcome 2 (CO2):

1. Explain how depth buffer method can be used to find out visible surfaces in a scene?.
2. Explain how scan line and back face detection algorithms can be used to find out visible surfaces in a scene?.
3. Difference between object and image space method.

Course Outcome 3 (CO3):

1. Assume that as an animator you are asked to develop a "card" using any 2D image editing tool which has the following effects : text effects, transformation effects, Place an image in the given text. Explain the above steps with suitable diagram.
2. Illustrate how to create a running tiger from moving automobile using morphing effect.
3. Assume that as an animator you are asked to develop an advertisement using any 2D animation tool with the following effects:
   i) Masking ii) Rotation iii) shape tweening iv) path tweening
   Explain the steps for the above effects with suitable diagram.
4. Discuss how to use different technology of "Motion Capture" in animated film?.

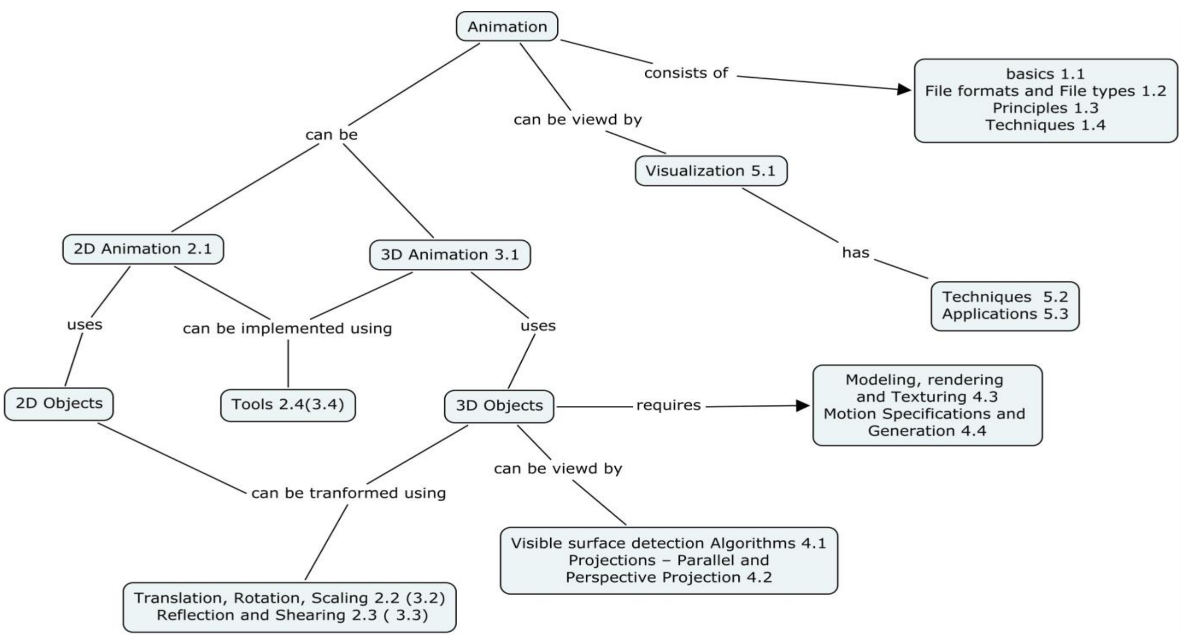
Course Outcome 4 (CO4)

1. With your own example or case study explain the practical steps that are needed for "good visualization"?.
2. Explain the applications of "Visualization" with an example.
3. List out the characteristics of data.
4. List out the techniques of visualization.

Course Outcome 5 (CO5)

1. Differentiate parallel projection from perspective projection.
2. Explain different types of parallel projection with necessary diagram and its corresponding transformation matrix.
3. Explain about Computer Graphics pipeline.

Concept Map



Syllabus

Introduction to animation: Animation basics, File formats, File types Color Models, Principles of animation, Techniques of animation – Traditional animation, stop motion and computer, Introduction to 2D and 3D animation. **2D ANIMATION-** Types of 2D Graphics, 2D graphics Techniques- Translation, Rotation, Scaling, Reflection and Shearing, 2D Animation Tools Introduction, Applications of 2D animation. **3D ANIMATION -** Phases of 3D Graphics - Modeling, Animation and Rendering, 3D graphics Techniques- Translation, Rotation, Scaling, Reflection and Shearing, 3D Animation Tools Introduction, Applications of 3D animation. **3D Viewing and Object Representation -** Visible surface detection (Back-Face Detection

Algorithm, Depth Buffer Method, Scan line Method) Algorithms, Projections – Parallel and Perspective Projection, 3D Requirements- Modeling, rendering, Texturing, Motion Capture and its Technology, Motion Specifications and Generation. **Visualization -** Visualization- Basics, Principles and Techniques of Visualization, Applications of Visualization, CASE Study.

Reference Books

1. Steve Roberts, 'Character Animation: 2D Skills for Better 3D',Second Edition, Focal Press, 2007.
2. Rick Parent, 'Computer Animation: Algorithms and Techniques', Third Edition, Elsevier, 2012.
3. Park, John Edgar, 'Understanding 3D Animation Using Maya', eighth Edition, Springer Publications, 2005.
4. Isaac Kerlow, 'The Art of 3D Computer Animation and Effects', 4th edition, Wiley Publications, 2009.
5. Zhigand Xiang, Roy Plastock: Theory and problems of Computer Graphics, Schaum's outline Series, Tata Mc-Graw hill edition. 2005.
6. Foley, James D Dam, Andries Van: Computer Graphics Principles and Practice, Pearson Education, 2002.
7. Donald Hearn and M. Pauline Baker: Computer Graphics: C Version, Pearson Education, Second Edition, 2006.

Course Contents and Lecture Schedule

No	Topic	No of Lectures
1.	**INTRODUCTION TO ANIMATION**	
1.1	Animation basics	1
1.2	File formats, File types and Color Models	1
1.3	Principles of animation	1
1.4	Techniques of animation – Traditional animation, stop motion and computer	1
1.5	Introduction to 2D and 3D animation	1
2	**2D ANIMATION**	

2.1	Types of 2D Graphics	1
2.2	2D graphics Techniques- Translation, Rotation, Scaling	1
2.3	2D graphics Techniques - Reflection and Shearing	3
2.4	2D Animation Tools Introduction	2
2.5	Applications of 2D animation	1
3.	**3D ANIMATION**	
3.1	Phases of 3D Graphics - Modeling, Animation and Rendering	1
3.2	3D graphics Techniques- Translation, Rotation, Scaling	3
3.3	3D graphics Techniques - Reflection and Shearing	2
3.4	3D Animation Tools Introduction	3
3.5	Applications of 3D animation	2
4	**3D Viewing and Object Representation**	
4.1	Visible surface detection (Back-Face Detection Algorithm, Depth Buffer Method, Scan line Method) Algorithms	2
4.2	Projections – Parallel and Perspective Projection	2
4.3	3D Requirements- Modeling, rendering, Texturing	2
4.4	Motion Capture and its Technology, Motion Specifications and Generation	2
5	**VISUALIZATION**	
5.1	Visualization- Basics	1
5.2	Principles and Techniques of Visualization	1
5.3	Applications of Visualization	1
5.4	CASE Study	1
	Total	**36**

Course Designer:

1. Dr. S.Sridevi sridevi@tce.edu

14CSGD0	ESSENTIALS OF INFORMATION TECHNOLOGY	Category	L	T	P	Credit
		GE	3	0	0	3

Preamble

This subject needs a pre-requisite of basic knowledge of computer organization and programming.

Course Outcomes

On the successful completion of the course, students will be able to

Explain the Hardware and software composition of a computer (CO1)	Understand
To understand the software life cycle and to overcome the complexity of real world applications through appropriate software model. (CO2)	Understand
Perform analysis on the real world complex application and identify its sub-systems by making use of ER models. (CO3)	Apply
Measure efficiency of a program using asymptotic notations. (CO4)	Apply
Design algorithms to sort data and to traverse a graph using various techniques. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S											
CO2.	S	M										
CO3.	S	M	M	L								
CO4.	S	M	M	L	L				L			L
CO5.	S	M	M	L	L				L			L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	40	20	30	30
Understand	30	40	20	20
Apply	30	40	50	50
Analyse	-	-	-	-
Evaluate	-	-	-	-

Create	-	-	-	-

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List out the essential constituents of a computer system.
2. Give an example for tertiary storage device.
3. Compare the speed, size and cost factor of primary and secondary storage device.
4. Write down steps that take place in performing an I/O operation.
5. List down the drawbacks of traditional file processing system.

Course Outcome 2 (CO2):

1. Differentiate procedure oriented and non-procedure oriented programming.
2. Describe the phases in software development life cycle.
3. Define software testing.
4. List down the various steps in debugging software.
5. Differentiate do-while and while control structures.

Course Outcome 3 (CO3):

1. What is the purpose of DBMS.
2. Consider a banking enterprise having an entity Payment. How it can be represented in ER Model.
3. Write a query to create two tables in SQL in such way that there is referencing between the tables.
4. Design an ER diagram for an University
5. Explain the various Constraints in Querying the Database.

Course Outcome 4 (CO4):

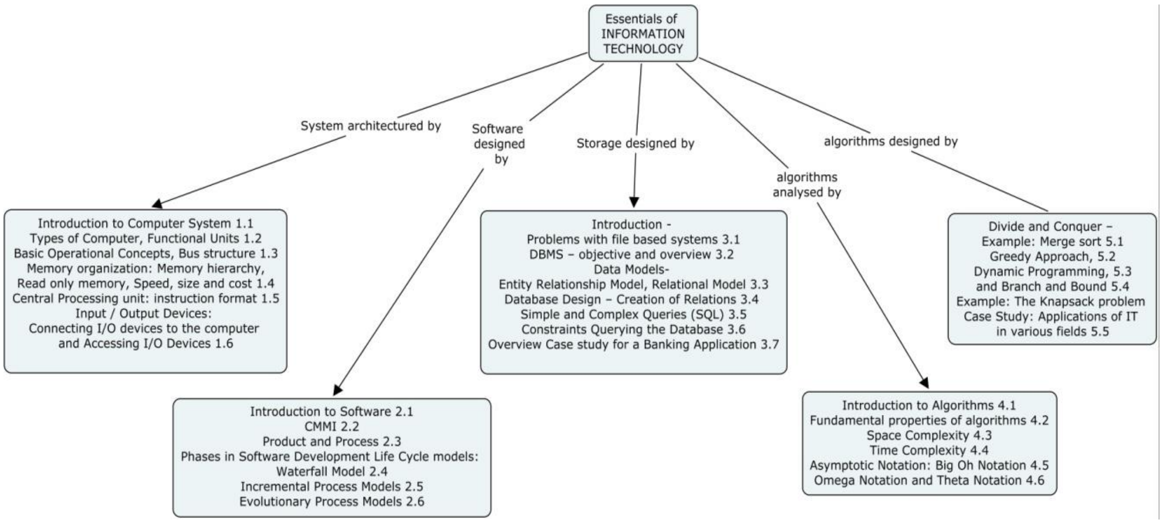
1. Define Principle of Optimality.
2. List down the four different asymptotic notations?
3. How is the running time of an algorithm calculated?
4. Give a solution for 8 Queens problem.
5. Differentiate time and space complexity.

Course Outcome 5 (CO5):

1. State the need for networking.
2. Differentiate LAN and WAN.
3. Write short notes on Internetworking devices.
4. Describe the issues involved in data networks.

5. List out the various wired network protocols.

Concept Map



Syllabus

Computer Hardware and System Software: Introduction to Computer System- Types of Computer- Functional Units, Basic Operational Concepts, Bus structure - Central Processing unit: instruction format - Memory organization: Memory hierarchy, Read only memory, Speed, size and cost - Input / Output Devices: Connecting I/O devices to the computer and Accessing I/O Devices.

Software development Life Cycle – Introduction to Software – CMMI - Product and Process - Phases in Software Development Life Cycle models: Waterfall Model, Incremental Process Models and Evolutionary Process Models.

Relational Data base management systems -Introduction – Problems with file based systems – DBMS – objective and overview- Data Models – Entity Relationship Model – Relational Model-Database Design – Creation of Relations - Simple and Complex Queries (SQL)– Constraints Querying the Database - Overview Case study for a Banking Application.

Design and Analysis of algorithms - Introduction to Algorithms- Fundamental properties of algorithms – Space Complexity - Time Complexity - Asymptotic Notations - Performance Measurement. Divide and Conquer – Merge sort.

Networks - Introduction to Networking-Introduction to Local Area Networks-Introduction to Wide Area Networks-Introduction to TCP/IP-Internetworking Devices-Internet Technologies-Implementation of Data Networks-Protocols Analysis-Network Design.

Text Books

1. Computer Organization & Architecture (Sixth Edition) By William Stallings (PHI)
2. M. Morris Mano, Computer System Architetutre, PHI 2003

3. Roger S. Pressman , Software Engineering: A Practitioner's Approach (5th edition), Mc Graw Hill 2000[Chapters 1,2,17,18]
4. Henry F Korth, Abraham Silberschatz, Database System Concept, 5th ed. McGraw-Hill International editions.
5. Ellis Horowitz ,Sartaj Sahni, Fundamentals of Computer Algorithms, Galgotia Publications,2002.
6. Andrew S. Tanenbaum, Computer Networks, 5th edition, Pearson Education

Course Contents and Lecture Schedule

Module No	Topic	No. of Lectures
1	**Computer Hardware and System Software (6)**	
Module No	Topic	No. of Lectures
1.1	Introduction to Computer System	1
1.2	Types of Computer, Functional Units	1
1.3	Basic Operational Concepts, Bus structure	1
1.4	Memory organization: Memory hierarchy, Read only memory, Speed, size and cost	1
1.5	Central Processing unit: Instruction format	1
1.6	Input / Output Devices: Connecting I/O devices to the computer and Accessing I/O Devices	1
2	**Software development Life Cycle (9)**	
2.1	Introduction to Software	1
2.2	CMMI	1
2.3	Product and Process	1
2.4	Phases in Software Development Life Cycle models: Waterfall Model	2
2.5	Incremental Process Models	2
2.6	Evolutionary Process Models	2

3	**Relational Data base management systems (9)**	
3.1	Introduction - Problems with file based systems	1
3.2	DBMS – objective and overview	1
3.3	Data Models- Entity Relationship Model, Relational Model	1
3.4	Database Design – Creation of Relations	1
3.5	Simple and Complex Queries (SQL)	2
3.6	Constraints Querying the Database	1
3.7	Overview Case study for a Banking Application	2
4	**Design and Analysis of algorithms (7)**	
4.1	Introduction to Algorithms	1
4.2	Space Complexity	1
4.3	Time Complexity	1
4.4	Asymptotic Notations	1
4.5	Performance Measurement	1
4.6	Divide and Conquer	1
4.7	Merge sort	1
5	**Networks (5)**	
5.1	Introduction to Networking-LAN,WAN	1
5.2	Introduction to TCP/IP	1
5.3	Internetworking Devices-Internet Technologies	1
5.4	Implementation of Data networks, Protocol Analysis and Network Design	2
	Total No of Hours	**36**

Course Designers:

1. Mr. S. Prasanna sprcse@tce.edu

14CSGE0	OBJECT ORIENTED CONCEPTS AND DESIGN	Category	L	T	P	Credit
		GE	3	0	0	3

Preamble

This course aims at facilitating the student to explore and understand the basics of object-oriented concepts and its technology. Perform analysis for the given complex system and model the object-oriented system for the given requirements by making use of UML diagrams.

Prerequisite

Basic of structured programming like C language.

Course Outcomes

On the successful completion of the course, students will be able to

Explain the importance of object-oriented model beyond conventional model (CO1)	Understand
To understand the object-oriented concepts and to overcome the complexity of real world applications through object oriented model. (CO2)	Understand
Perform analysis on the real world complex application and identify its sub-systems by making use of modularity concepts. (CO3)	Apply
Model the static nature of the real world system through UML diagrams. (CO4)	Apply
Model the dynamic nature of the real world system through UML diagrams. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	S											
CO2.	S	M										
CO3.	S	M	M	L								
CO4.	S	M	M	L	L				L			L
CO5.	S	M	M	L	L				L			L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	10
Understand	50	20	20	30
Apply	30	60	60	60

Analyse	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the benefits of OOD?
2. Summarize the difference between conventional programming and object oriented programming.
3. Define Object Oriented Analysis.
4. What is modelling? Write the various models in Object Oriented Development.
5. Differentiate object oriented and object based technology.

Course Outcome 2 (CO2):

1. Explain in detail the various elements of Object Model
2. Explain the evolution of object oriented model in detail.
3. List out the Properties of simple and complex software systems.
4. Outline how to address the complexity of a complex system.
5. Write about five attributes of a complex system

Course Outcome 3 (CO3):

1. Explain the different kinds of decomposition with suitable example.
2. For the Library Management system, identify the relationship between the modules/sub systems and give the OO model.
3. Write a note on “ Parameterized class”.
4. What is the purpose of identifying the semantics of classes and objects?
5. How will you identify the quality classes? Explain.

Course Outcome 4 (CO4):

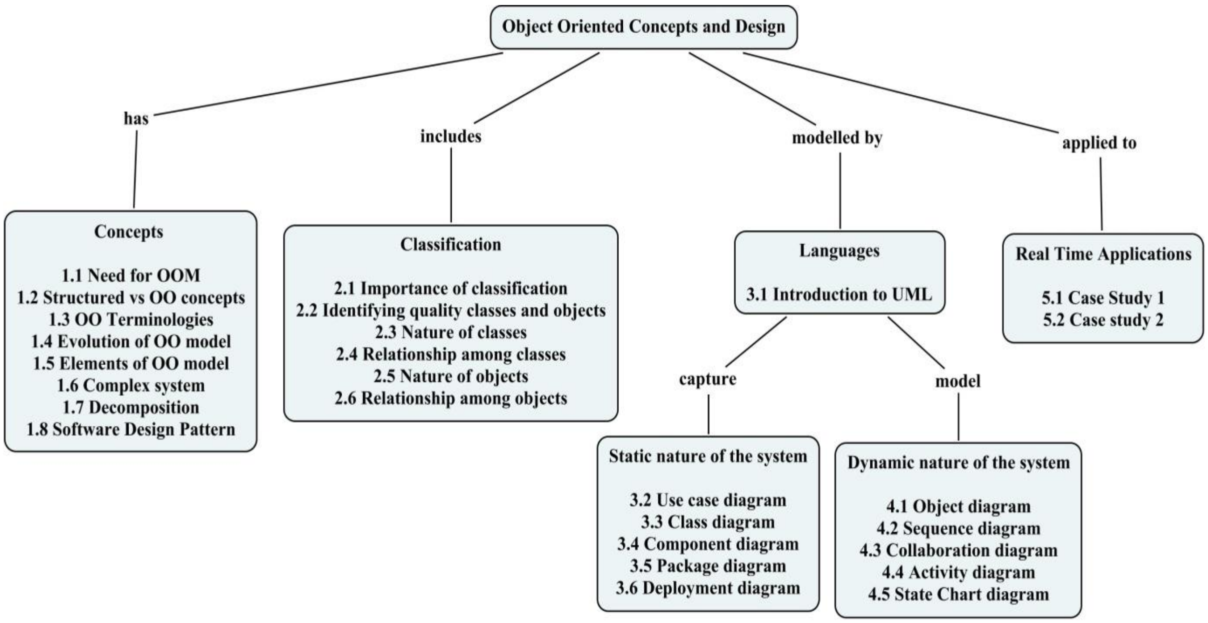
1. What is the role of classes and objects in analysis and design?
2. How will a Deployment diagram be helpful in the development process?
3. Demonstrate the difference between the Composite and Simple aggregations with an example.
4. Model the static structure of the Car Company scenario which will be helpful for the implementer to code the system.
5. What is multiplicity? Mention its use.

Course Outcome 5 (CO5):

1. Compare and contrast the Sequence and Collaboration diagrams.
2. Define Activity diagram.
3. List the use of Fork and Join in Activity diagram. How are they represented?

4. Describe the state diagram with your own example.
5. Construct the State Chart diagram for the given Car Company System. Make use of all the possible adornment of a State diagram.

Concept Map



Syllabus

Introduction : Need for Object Oriented Model – Structured vs Object Oriented concepts - Object oriented concepts and terminologies - Evolution of Object model - Elements of object model - Complexity of the software system –- Bringing order to chaos – Decomposing the system - Applying the object model – Software Design Patterns. **Classification:** Importance of proper classifications – Identifying quality classes and objects – Nature of classes – Relationship among classes - Nature of objects – Relationship among objects - **Unified Modeling Language: Static Diagrams –** UML introduction – Use case diagram – Class diagram - Component diagram – Package and Deployment diagrams. **Unified Modeling Language: Dynamic Diagrams -** Object diagram – Sequence diagram **-** Collaboration diagram **-** Activity diagram - Statechart diagram**. Object Oriented Design Application –** Case studies.

Reference Books

1. Grady Booch , Robert A. Maksimchuk , Michael W. Engle, Bobbi J. Young, Jim Conallen , Kelli A. Houston , " Object-Oriented Analysis and Design with Applications", Addison-Wesley Professional, 3rd Edition, 2007.
2. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modelling Language – User Guide", Addison-Wesley,nd Edition, 2005.

3. Ali Bahrami, ”Object Oriented System development “, McGraw-Hill international edition 1999.
4. John Deacon, “Object-Oriented Analysis and Design”, Addison-Wesley, First Edition 2005.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**Introduction (8)**	
1.1	Need for Object Oriented Model	1
1.2	Structured vs Object Oriented concepts	1
1.3	Object oriented concepts and terminologies	1
1.4	Evolution of Object model	1
1.5	Elements of object model	1
1.6	Complexity of the software system, Bringing order to chaos	1
1.7	Decomposing the system, Applying the object model.	1
1.8	Software Design Patterns	1
2	**Classification (8)**	
2.1	Importance of proper classifications	1
2.2	Identifying quality classes and objects	1
2.3	Nature of classes	1
2.4	Relationship among classes	2
2.5	Nature of objects	1
2.6	Relationship among objects	2
3	**Unified Modeling Language – Static Diagrams (8)**	
3.1	UML introduction	1
3.2	Use case diagram	2
3.3	Class diagram	2
3.4	Component diagram	1
3.5	Package diagram	1
3.6	Deployment diagrams.	1
4	**Unified Modeling Language – Dynamic Diagrams (6)**	
4.1	Object diagram	1
4.2	Sequence diagram	1
4.3	Collaboration diagram	1
4.4	Activity diagram	2

Module No.	Topic	No. of Lectures
4.5	State chart diagram	1
5	**OOD Application (6)**	
5.1	Case Study-1	3
5.2	Case Study-1	3
	Total	**36**

Course Designers:

1. Mrs. A.M.Rajeswari amrcse@tce.edu

14CSGF0	ENTERPRISE APPLICATION DEVELOPMENT	Category	L	T	P	Credit
		GE	3	0	0	3

Preamble

This course introduces basic building blocks of .NET framework and development of windows and web based applications on .NET using C# language. The students are introduced with various constructs of C# language using which they can develop applications on the .NET framework. The idea is to explore the features of C# and .NET framework and to use it for application development.

Prerequisite

Basics of Object oriented programming.

Course Outcomes

On the successful completion of the course, students will be able to

Explain the various building blocks of .NET framework and the process of compiling the C# source codes. (CO1)	Understand
Write programs using C# basic constructs (CO2)	Apply
Implement object oriented concepts using C# for developing an application. (CO3)	Apply
Retrieve data using ADO .NET and Develop windows application based on the given requirements. (CO4)	Apply
Build Web applications using web forms and web services. (CO5)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	L				L							
CO2.	M	L			M							
CO3.	M	L	L		M							
CO4.	S	M	M		M				L			L
CO5.	S	M	M		M				L			L

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	30	20	20	20
Understand	40	40	30	40
Apply	30	40	50	40
Analyse	0	0	0	0
Evaluate	0	0	0	0
Create	0	0	0	0

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the silent features of .NET framework.
2. Explain the use of .NET assemblies.
3. What is the use of .NET framework SDK?

Course Outcome 2 (CO2):

1. Explain with example enumerations and arrays in C#.
2. Compare value type and reference type.
3. Write a C# code to compare two strings using "Equals" method.

Course Outcome 3 (CO3):

1. How does C# support multiple inheritance?
2. What do you mean by delegates? State their use with an example.
3. Consider a student class with feet and inches as attributes which describes the height of the student. Write a C# program to overload the + operator and to find the average of N students.
4. Explain user defined exceptions with an example.

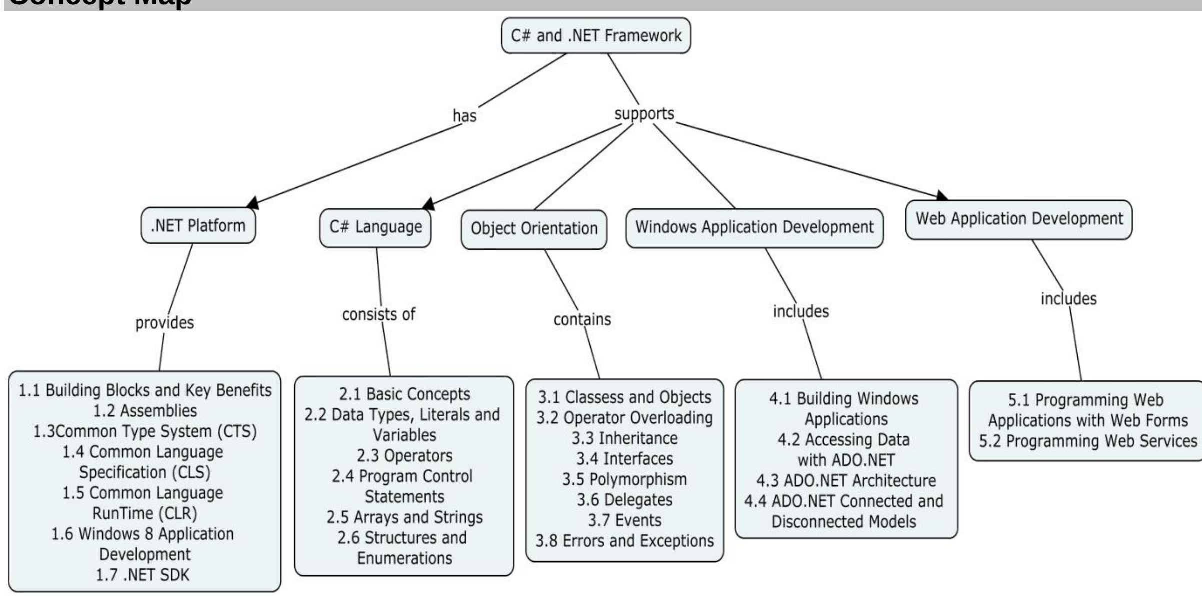
Course Outcome 4 (CO4):

1. How does ADO.NET connected and disconnected models differ from each other?
2. Explain the process of creating menus in a windows-based application.
3. Write a window based application to display a message

Course Outcome 5 (CO5):

1. Develop a web application for railway ticket booking and cancellation process with following features
    - User friendly interfaces
    - Implementation of transaction properties
    - Handling of exceptions.
2. Define a web service? List few real-time web services.
3. Apply the features of ASP.NET web forms and web pages and design a pay roll application.

Concept Map



Syllabus

Introducing C# and .NET Platform - Key Benefits of the .NET Platform, Building Blocks of the .NET Platform, An Overview of .NET Assemblies, Understanding the Common Type System, Understanding the Common Language Specification, Understanding the Common Language Runtime, Building Windows 8 Applications, Role of .NET Under Windows 8. **Building C# Applications -** Role of the .NET Framework 4.5 SDK, Building .NET Applications Using Visual C# Express.
Introduction to C# - Overview of C#, Data Types, Literals, and Variables, Operators, Program Control Statements, Arrays and Strings, Structures and Enumerations.
Object oriented aspects of C# - Introducing Classes and Objects, Operator Overloading, Inheritance, Interfaces, Polymorphism, Delegates, Events, Errors and Exception Handling.
Application Development on .NET - Building Windows Applications, Accessing Data with ADO.NET, ADO.NET Architecture , ADO.NET Connected and Disconnected Models
Web based Application Development on .NET - Programming Web Applications with Web Forms, Programming Web Services.

Text Books

1. Andrew Troelsen, Pro C# 5.0 and the .NET 4.5 Framework, Sixth Edition, Apress, 2012.
2. Herbert Schildt, C# 4.0 - The Complete Reference, First Edition, Tata McGraw-Hill, 2012.
3. J. Liberty, Programming C#", Fourth Edition, O'Reilly, 2005.

Reference Books

1. Ian Griffiths, Matthew Adams and Jesse Liberty, Programming C# 4.0, Sixth Edition, O'Reilly, 2010.
2. Christian Nagel et al, Professional C# 2012 with .NET 4.5, Wiley India, 2012.
3. E.Balagurusamy, Programming in C# - A Primer, Third Edition,Tata McGraw Hill, 2010.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1.	**Introducing C# and .NET Platform (7)**	
1.1	Key Benefits of the .NET Platform, Building Blocks of the .NET Platform	1
1.2	An Overview of .NET Assemblies	1
1.3	Understanding the Common Type System	1
1.4	Understanding the Common Language Specification	1
1.5	Understanding the Common Language Runtime	1
1.6	Building Windows 8 Applications, Role of .NET Under Windows 8	1
1.7	**Building C# Applications -** Role of the .NET Framework 4.5 SDK, Building .NET Applications Using Visual C# Express	1
2.	**Introduction to C# (7)**	
2.1	Overview of C#	1
2.2	Data Types, Literals, and Variables	1
2.3	Operators	1
2.4	Program Control Statements	1
2.5	Arrays and Strings	2
2.6	Structures and Enumerations	1
3.	**Object oriented aspects of C# (8)**	
3.1	Introducing Classes and Objects	1

Module No.	Topic	No. of Lectures
3.2	Operator Overloading	1
3.3	Inheritance	1
3.4	Interfaces	1
3.5	Polymorphism	1
3.6	Delegates	1
3.7	Events	1
3.8	Errors and Exception Handling	1
4.	**Application Development on .NET (7)**	
4.1	Building Windows Applications	3
4.2	Accessing Data with ADO.NET	1
4.3	ADO.NET Architecture	1
4.4	ADO.NET Connected and Disconnected Models	2
5.	**Web based Application Development on .NET (7)**	
5.1	Programming Web Applications with Web Forms	4
5.2	Programming Web Services	3
	Total No. of hours	**36**

Course Designer:

1. Mrs. B.Subbulakshmi bscse@tce.edu

		Category	L	T	P	Credit
14CSGG0	**PROGRAMMING USING PYTHON**	PE	3	0	0	3

Preamble

The syllabus is designed to understand the key features of the programming Language Python. The main focus is on the execution principle of object concepts. Python Programming in both Object Oriented and Structured Programming concepts are analyzed in order to get the extract of this programming paradigm. More than that, Software application is done using this programming language.

Prerequisite

NIL

Course Outcomes

On the successful completion of the course, students will be able to

Identify the suitable object type for the given input. (CO1)	Understand
Apply the concepts of conditional statements in solving simple mathematical problems like Grading Systems.(CO2)	Apply
Apply the concepts of loops and decision statements in solving simple mathematical problems.(CO3)	Apply
Apply the concepts of functions and Argument Passing in real world problems like Towers of Hanoi. (CO4)	Apply
Apply the concepts of Object Oriented Programming with Structured programming and analyze its performance in terms of Code Reusability and Security. (CO5)	Apply
Design applications for business problems such as Payroll Management, Library Management etc. (CO6)	Apply

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1.	M	M	M	S	M	S	S		M	S	M	M
CO2.	S	M	M	S	M	S	S		M	S	M	M
CO3.	S	M	M	S	M	S	S		M	S	M	M
CO4.	S	M	M	S	M	S	S		M	S	M	M
CO5.	S	M	M	S	M	S	S		M	S	M	M
CO6.	S	M	M	S	M	S	S		M	S	M	M

S- Strong; M-Medium; L-Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests			Terminal Examination
	1	2	3	
Remember	20	20	20	20
Understand	40	40	30	30
Apply	40	40	50	50
Analyze	-	-	-	-
Evaluate	-	-	-	-
Create	-	-	-	-

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Name four of Python's core data types. (Remember)
2. Why are they called "core" data types? (Understand)
3. What does "immutable" mean, and which three of Python's core types are considered immutable? (Understand)
4. What does "sequence" mean, and which three types fall into that category? (Understand)

Course Outcome 2 (CO2):

1. What do the words True and False mean? (Understand)
2. How might you code a multiway branch in Python?. Explain with an example. (Apply)
3. How can you code an if/else statement as an expression in Python for checking a prime number? (Apply)
4. How can you make a single statement span many lines?Explain with an example. (Apply)

Course Outcome 3 (CO3)

1. What can a range be used for in a for loop? (Remember)
2. What are the main functional differences between a while and a for? (Understand)
3. What's the difference between break and continue? (Understand)
4. When is a loop's else clause executed? (Understand)
5. How can you code a counter-based loop in finding the sum of n numbers using Python? (Apply)

Course Outcome 4 (CO4)

1. What is the point of coding functions? (Understand)

2. What does a function return if it has no return statement in it? (Understand)
3. Write a Python Program to find the factorial of a number using functions. (Apply)
4. What Is The Output Of The Following Code Snippet? (Apply)

```
def myfunc(text, num):
    while num > 0:
        print(text)
     num = num - 1
myfunc('Hello', 4)
```

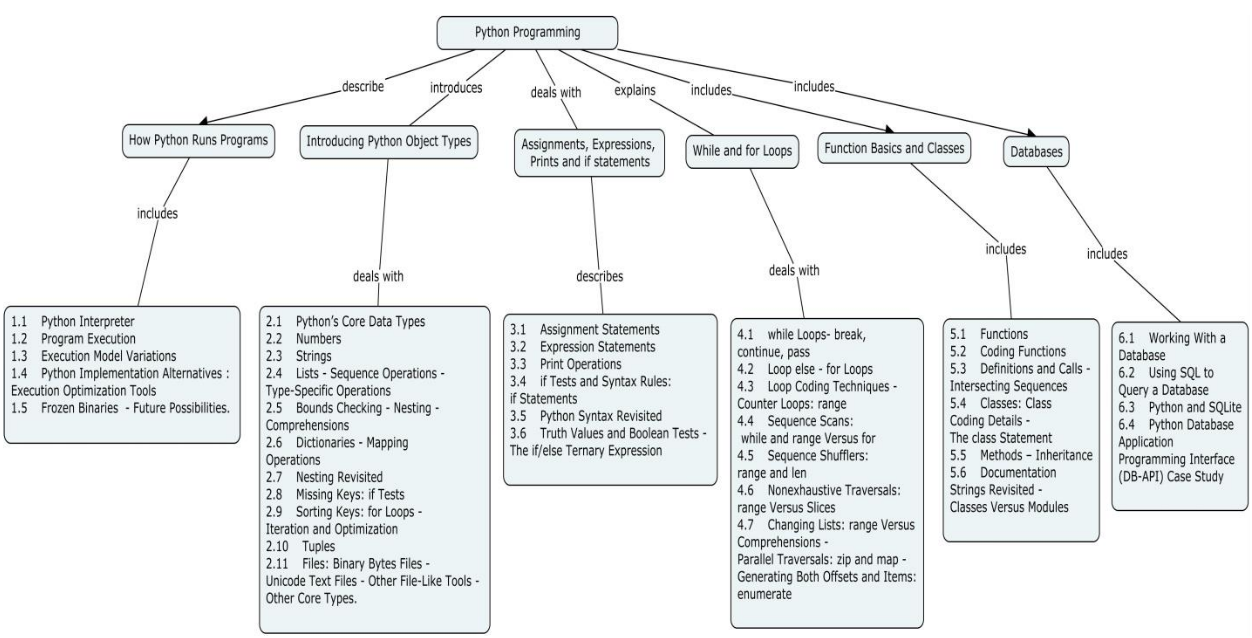
Course Outcome 5 (CO5)

1. How are instances and classes created? (Remember)
2. What does self mean in a Python class? (Remember)
3. How are classes related to modules? (Understand)
4. What are two key concepts required to understand Python OOP code? (Understand)
5. Write a Python program for Student Record Management with and without using OOP concepts and compare both for Code Reusability and Security. (Apply)

Course Outcome 6 (CO6)

1. Develop a mini project on Pay Roll Management using Python (Apply)
2. Develop a mini project on Library Management using Python (Apply)

Concept Map



Syllabus

How Python Runs Programs: Python Interpreter - Program Execution - Execution Model Variations - Python Implementation Alternatives : Execution Optimization Tools - Frozen Binaries - Future Possibilities. **Python Object Types:** Python's Core Data Types - Numbers - Strings - Lists - Sequence Operations - Type-Specific Operations. Bounds Checking - Nesting - Comprehensions. Dictionaries - Mapping Operations - Nesting Revisited - Missing Keys: if Tests - Sorting Keys: for Loops - Iteration and Optimization. Tuples ,Files: Binary Bytes Files - Unicode Text Files - Other File-Like Tools - Other Core Types. **Assignments, Expressions, Prints and if statements:** Assignment Statements - Expression Statements - Print Operations. if Tests and Syntax Rules: if Statements - Python Syntax Revisited - Truth Values and Boolean Tests - The if/else Ternary Expression. **While and for Loops:** while Loops - break, continue, pass, and the Loop else - for Loops - Loop Coding Techniques -Counter Loops: range - Sequence Scans: while and range Versus for - Sequence Shufflers: range and len - Nonexhaustive Traversals: range Versus Slices - Changing Lists: range Versus Comprehensions - Parallel Traversals: zip and map - Generating Both Offsets and Items: enumerate. **Function Basics and Classes:** Functions - Coding Functions - Definitions and Calls - Intersecting Sequences – Classes: Class Coding Details - The class Statement - Methods - Inheritance - Documentation Strings Revisited - Classes Versus Modules.**Databases:** Working With a Database- Using SQL to Query a Database-Python and SQLite- Python Database Application Programming Interface (DB-API) Case Study

Reference Books

1. Mark Lutz, "Powerful Object Oriented Programming Learning Python" Oreilly, 5th Edition, 2013.
2. Mark Summerfield, "Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education,2008.
3. Zed Shaw, "Learn Python the Hard Way" Addison-Wesley, 2013.
4. Martin C. Brown , "Python: The Complete Reference ",McGraw Hill,2018.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Lectures
1	**How Python Runs Programs:**	
1.1	Python Interpreter	1
1.2	Program Execution	
1.3	Execution Model Variations	1
1.4	Python Implementation Alternatives : Execution Optimization Tools	1
1.5	Frozen Binaries - Future Possibilities.	1
2	**Introducing Python Object Types**	
2.1	Python's Core Data Types	1

2.2	Numbers	1
2.3	Strings	1
2.4	Lists - Sequence Operations - Type-Specific Operations	1
2.5	Bounds Checking - Nesting - Comprehensions	1
2.6	Dictionaries - Mapping Operations	1
2.7	Nesting Revisited	1
2.8	Missing Keys: if Tests	1
2.9	Sorting Keys: for Loops - Iteration and Optimization	
2.10	Tuples	1
2.11	Files: Binary Bytes Files - Unicode Text Files - Other File-Like Tools - Other Core Types.	1
3	**Assignments, Expressions, Prints and if statements**.	
3.1	Assignment Statements	1
3.2	Expression Statements	1
3.3	Print Operations	1
3.4	if Tests and Syntax Rules: if Statements	1
3.5	Python Syntax Revisited	1
3.6	Truth Values and Boolean Tests - The if/else Ternary Expression	1
4	**While and for Loops**	
4.1	while Loops- break, continue, pass	1
4.2	Loop else - for Loops	1
4.3	Loop Coding Techniques -Counter Loops: range	1
4.4	Sequence Scans: while and range Versus for	1
4.5	Sequence Shufflers: range and len	1
4.6	Nonexhaustive Traversals: range Versus Slices	1
4.7	Changing Lists: range Versus Comprehensions - Parallel Traversals: zip and map - Generating Both Offsets and Items: enumerate	1
5	**Function Basics and Classes**	
5.1	Functions	1
5.2	Coding Functions	1
5.3	Definitions and Calls - Intersecting Sequences	1
5.4	Classes: Class Coding Details - The class Statement	1
5.5	Methods – Inheritance	1
5.6	Documentation Strings Revisited - Classes Versus Modules	1
6	**Databases**	
6.1	Working With a Database	1

6.2	Using SQL to Query a Database	1
6.3	Python and SQLite	
6.4	Python Database Application Programming Interface (DB-API) Case Study	1
	Total hours	36

Course Designers:

1. Ms Raja Lavanya rlit@tce.edu
2. Ms. Thiraviaselvi G gts@tce.edu