Department of Computer Science and Engineering

B.E. Degree (CSE Program)

Curriculum for the Students who are admitted for the Academic year 2013 – 2014

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

Department Electives

Sl.No	Subject Code	Subject Name
1.	CCA	Numerical Methods and Number Theory
2.	CCB	Information Retrieval
3.	CCC	Data Mining Concepts and Techniques
4.	CCD	Agent Based System
5.	CCE	Wireless Networks
6.	CCF	Grid Computing
7.	CCG	Modeling and Simulation
8.	CCH	Parallel Computing
9.	CCJ	Service Oriented Architecture
10.	CCK	Software Risk Assessment and Management
11.	CCL	Data warehousing
12.	CCM	Enterprise Project Development using FOSS
13.	CCN	Storage Infrastructure Management

One Credit Courses

Sl.No	Subject Code	Subject Name
1.	C1A	Rich Internet Applications
2.	C1B	Advanced Operating Systems
3.	C1C	Mobile Applications Development
4.	C1D	Basics of Application Security
5.	C1E	Cyber Security
6.	C1F	Green Data Center
7.	C1G	Introduction to Avionics
8.	C1H	Introduction to IT Service Management
9.	C1I	Introduction to Mainframe Systems
10.	C1J	Practical approaches to Networking
11.	C1K	Introduction to Microsoft Active Directory Domain Services

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

SECOND SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 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

Board of Studies Meeting Approved on 13.10.2012 Approved in 45th Academic Council Meeting on 24.11.2012

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the academic year 2013 – 2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014)

SECOND SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
C 21	Engineering Mathematics II	BS	4	-	-	4
C 22	Digital Systems	DC	3	1	-	4
C 23	Computer Organization	DC	3	-	-	3
C 24	Procedural Programming	ES	3	-	-	3
C 25	Data Structures	ES	3	-	-	3
C 26	Environment and Ecology	HSS	2	-	-	2
PRACTICAL						
C 27	Data structures Lab	DC	-	-	3	1
C 28	Procedural Programming Lab	ES	-	-	3	1
C 29	Workshop	ES	-	-	3	1
Total			**18**	**1**	**9**	**22**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
DC : Department Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS

(For the candidates admitted from 2013-2014)

SECOND SEMESTER

S.No.	Sub. code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C21	Engineering Mathematics II	3	50	50	100	25	50
2	C 22	Digital Systems	3	50	50	100	25	50
3	C 23	Computer Organization	3	50	50	100	25	50
4	C 24	Procedural Programming	3	50	50	100	25	50
5	C25	Data Structures	3	50	50	100	25	50
6	C26	Environment and Ecology	3	50	50	100	25	50
PRACTICAL								
7	C 27	Data structures Lab	3	50	50	100	25	50
8	C 28	Procedural Programming Lab	3	50	50	100	25	50
9	C29	Workshop	3	50	50	100	25	50

* CA evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub Code	Lectures	Tutorial	Practical	Credit
C 21	4	-	-	4

C21 Engineering Mathematics II 4:0

(Common to all branches of Engineering B21, C21, D21, E21, G21, T21)

Program Outcomes addressed

a. An ability to apply knowledge of engineering, information technology, mathematics, and science
b. An ability to identify, formulate and solve engineering problems
c. An ability to engage in life-long learning

Competencies: At the end of the course the students should be able to

1. Formulate and solve problems of engineering dynamics using different differential operators.
2. Formulate the problem of computing areas and volumes through vector integration, and determine them by applying Green, Stokes and Divergence theorems
3. Determine maxim and minima of functions of several variables using analytical and Lagrangian multipliers methods
4. Determine the values of multiple integrals directly or by changing the order of integration or by making transformation with Jacobians.
5. Determine areas and volumes of geometrical figures using multiple integrals, beta and gamma functions.
6. Analyze functions of complex variable in terms of continuity, differentiability and analyticity.
7. Apply Cauchy-Riemann equations and harmonic functions to problems of fluid mechanics, thermodynamics and electro-magnetic fields.
8. Find singularities of complex functions and determine the values of integrals using residues.
9. Geometrically interpret conformal and bilinear transformations

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level Learning Objectives

Understand

1. Distinguish between solenoidal and irrotational vectors?
2. How do you understand an analytic function? Give examples of an analytic function and a non-analytic function?
3. Distinguish among the three types of singularities of a complex valued function f(z)?
4. Examine the following function for extreme values

$$u = x^4 + y^4 - 2x^2 + 4xy - 2y^2$$

5. Explain the difference between a stationary value and an extreme value of a real valued function f(x, y) illustrating with examples?
6. If the three thermodynamic variables P, V and T are connected by a relation f(P,V,T) = 0, show that

$$\left(\frac{\partial P}{\partial T}\right)_V \left(\frac{\partial T}{\partial V}\right)_P \left(\frac{\partial V}{\partial P}\right)_T = -1$$

7. Verify Stoke's theorem for $\bar{F} = (2x - y)\bar{i} - yz^2\bar{j} - y^2z\bar{k}$ where S is the upper half of the sphere $z^2 + y^2 + z^2 = 1$ and C is the boundary in the xy plane?

Apply

1. Find the unit vector normal to the surface $x^2 + 2y^2 + z^2 = 7$ at (1, -1, 2)?
2. Establish the result

$$Div(\bar{u} \times \bar{v}) = \bar{v}.curl\,\bar{u} - u.curl\,\bar{v})$$

Deduce that if $\bar{u}, \bar{v}$ are irrotational, then $\bar{u} \times \bar{v}$ is solenoidal?

3. Transform the equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$ into polar coordinates?

4. A fluid motion is given by $\bar{V} = (y+z)\bar{i} + (z+x)\bar{j} + (x+y)\bar{k}$. Is this motion irrotational? If so, find the velocity potential? Is this motion possiblefor an incompressible fluid?

5. Determine the value of $\int_0^5 \int_0^{x^2} x(x^2 + y^2)\,dx\,dy$?

6. Find the volume bounded by the cylinder $x^2 + y^2 = 4$ and the planes $y + z = 4$ and $z = 0$ using multiple integrals?

7. The plane $\frac{x}{a} + \frac{y}{b} + \frac{z}{c} = 1$ meets the axes at A, B and C. Apply Dirichlet's integrals to find the volume of the tetrahedron OABC?

8. Transform the integral $\int_0^{\pi} \int_0^{a} r^3 \sin\theta \cos\theta \, dr \, d\theta$ into Cartesian form and determine the value of the integral?

9. If $\omega = \phi + i\psi$ represents the complex potential for an electric field and $\psi = x^2 - y^2 + \frac{x}{(x^2+y^2)}$, determine the function ϕ?

10. Find the bilinear transformation mapping the points z = 1, i, -1 into the points ω = 0, 1, ∞?

11. Use Cauchy's integral formula to determine $\int_C \frac{e^{2z}}{(z-1)(z-2)}$ where 'C' is the circle |z| = 3?

12. Obtain the Laurent series expansion of f(z) = $\frac{z}{(z^2-1)(z^2+4)}$ for regions (a) |z|<1 (b) 1<|z|<2 (c)|z|>2?

13. Determine the value of the integral $\int_{-\infty}^{\infty} \frac{x^2\,dx}{(x^2+1)(x^2+4)}$ using the theorem of residues?

14. Apply the method of contour integration to prove that $\int_0^{2\Pi} \frac{\cos 2\theta}{5 + 4\cos\theta} d\theta = \frac{\Pi}{6}$ clearly indicating the contour selected?

15. A tent on a square base of side x, has vertical sides of height y, and the top is a regular pyramid of height h. Find x and y in terms of h, if the canvas required for its construction is to be minimum for the tent to have a given capacity?

Course Contents:

1. Functions of Several Variables:

1.1 Partial derivatives and Jacobians

1.2 Total differentiation and applications

1.3 Lagrangian Multiplier method

1.4 Applications to Maxima and Minima

2. Multiple Integrals:

2.1 Double integrals and areas

2.2 Triple integrals and volumes

2.3 Change of order of integration

2.4 Beta and Gamma functions with applications

2.5 Change of variables between Cartesian and polar with applications

3. Vector calculus:

3.1 Vector Differentiation with simple applications

3.2 Operators Grad, div and curl with properties

3.3 Applications to Physics

3.4 Vector Integration(three famous theorems)

3.5 Applications to areas and volumes

4. Complex Differentiation:

4.1 Analytic functions, C-R equations and properties

4.2 Harmonic Functions and Milne-Thompson Method

4.3 Applications to flow problems

4.4Conformal maps and bilinear transformations

4.5 Applications of the bilinear transformations

5. Complex Integration:

5.1 Cauchy's theorem and consequences

5.2 Evaluating integrals using Cauchy's integral formula

5.3 Taylor and Laurent expansions

5.4 Singularities, poles and Cauchy residue theorem

5.5 Contour integration using unit circle and semicircular contours

Syllabus

Functions of Several Variables: Partial derivatives and Jacobians, Total differentiation and applications, Lagrangian Multiplier method, Applications to Maxima and Minima **Multiple Integrals:** Double integrals and areas, Triple integrals and volumes, Change of order of integration, Beta and Gamma functions with applications, Change of variables between Cartesian and polar with applications **Vector calculus:** Vector Differentiation with simple applications, Operators Grad, div and curl with properties, Applications to Physics, Vector Integration(three famous theorems), Applications to areas and volumes **Complex Differentiation:** Analytic functions, C-R equations and properties, Harmonic Functions and Milne-Thompson Method, Applications to flow problems, Conformal maps and bilinear transformations, Applications of the bilinear transformations **Complex Integration:** Cauchy's theorem and consequences, Evaluating integrals using Cauchy's integral formula, Taylor and Laurent expansions, Singularities, poles and Cauchy residue theorem, Contour integration using unit circle and semicircular contours

Text Book

B.S. Grewal: Higher Engineering Mathematics, 39th Edn., Khanna Publishers, New Delhi

References

1. Lecture Notes by the faculty of Department of Mathematics, TCE, Madurai
2. Veerarajan T.: Engineering Mathematics, 3rd Edn., Tata McGraw Hill, New Delhi, 2004
3. Venkataraman M.K.: Multiple Integrals and Gamma, Beta functions, National Publishing Co., 2004
4. Manish Goyal: Textbook of Engineering Mathematics, Lakshmi Publishing, New Delhi
5. Venkataraman M.K.: Engineering Mathematics (Vol. II), ational Publishing, Chennai
6. Kreyszig E.: Advanced Engineering Mathematics, 8th Edn. John Wiley & Sons

Lecture Schedule

No.	Topic	No. of Lectures
1.	**Functions of Several Variables**	
1.1	Partial derivatives and Jacobians	2
1.2	Total differentiation and applications	2

No.	Topic	No. of Lectures
1.3	Lagrangian Multiplier method	2
1.4	Applications to Maxima and Minima	2
2.	**Multiple Integrals**	
2.1	Double integrals and areas	1
2.2	Triple integrals and volumes	1
2.3	Change of order of integration	2
2.4	Beta and Gamma functions with applications	2
2.5	Change of Variables between Cartesian and Polar with applications	1
3	**Vector calculus**	
3.1	Vector Differentiation with simple applications	1
3.2	Operators Grad, div and curl with properties	3
3.3	Applications to Physics	1
3.4	Vector Integration(three famous theorems)	4
3.5	Applications to areas and volumes	3
4	**Complex Differentiation:**	
4.1	Analytic functions, C-R equations and properties	3
4.2	Harmonic Functions and Milne-Thompson Method	2
4.3	Applications to flow problems	1
4.4	Conformal maps and bilinear transformations	2
4.5	Applications of the bilinear transformations	2
5.	**Complex Integration**	
5.1	Cauchy's theorem and consequences	2
5.2	Evaluating integrals using Cauchy's integral	2

No.	Topic	No. of Lectures
	formula	
5.3	Taylor and Laurent expansions	2
5.4	Singularities, poles and Cauchy residue theorem	2
5.5	Contour integration using unit circle and semicircular contours	4

Course Designers:

1. **V. Mohan vmohan@tce.edu**
2. **N. Kannan nkmat@tce.edu**
3. **M. Kameswari mkmat@tce.edu**
4. **K. Angaleeswari kamat@tce.edu**
5. **P. Subramanian psmat@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 22	3	1	-	4

C22: Digital Systems 3:1

Preamble:

The proposed course is offered in the second semester. It will be helpful for the course system software as well as for computer architecture and organization. In the designing this course the following assumptions are made.

- The course aims at design of combinational and sequential functions at gate level.
- No reference will be made to any technology or logic family.
- No hardware aspects(voltages, currents, noise margin, transients etc.) other than delay time would be considered in designing logic function.
- As SSIs and MSIs are not made in use any more, no reference needs to be made to these
  ICs.

Program outcomes addressed

- Graduates will demonstrate knowledge of mathematics, science and engineering.
- Graduates will demonstrate an ability to design a system, components or process as per needs and specifications.
- Graduates will demonstrate skills to use modern engineering tools, software's and equipment to analyze problems.

Competencies:

1. Determine the functional behaviour and timing performance of a given combinational logic circuit.
2. Determine the functional behaviour and timing performance of given sequential logic circuit using state diagrams, timing diagrams, and PS-NS-O tables.
3. Determine the behaviour of a given digital circuit with regard to hazards, asynchronous inputs, and output races.
4. Design digital circuits to perform specified combinational and sequential functions.

Assessment Pattern

	Bloom's Category	Test 1	Test2	Test3	End-semester examination
1	Remember	20	20	20	20
2	Understand	20	20	20	20
3	Apply	60	30	30	30
4	Analyze	0	0	0	0
5	Evaluation	0	0	0	0
6	Create	0	30	30	30

Course Level Learning Objectives

Remember

1. State Demorgan's theorem.
2. What is meant by multiplexer?
3. What is BCD code?
4. What is priority encoder?
5. Write the truth table of SR flip flop.
6. What is modulus of a counter?

Understand

1. Differentiate between a BCD code and a binary number.
2. Differentiate between a truth table and an excitation table.
3. Differentiate between a combinational circuit and a sequential circuit.
4. Explain why JK flip flops produce more don't care terms than the other flop flop types, even when all of the states are specified.
5. Differentiate between ripple and synchronous counter.

Apply

1. Find all the prime Implicants for the following Boolean functions and determine which are essential.

   i)$F(w,x,y,z) = \sum(0,2,4,5,6,7,8,10,13,15)$

   ii)$F(A,B,C,D) = \sum(0,2,3,5,7,8,10,11,14,15)$

2. Reduce the number of states in the following state table, and tabulate the reduced state table:

Present state	Next state		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

3. Minimize the following Boolean function using Quine-McCluskey method.
   $F(x_1,x_2,x_3,x_4) = \Sigma(0,5,7,8,9,10,11,14,15)$
4. Simplify the following Boolean Expression using K Map and construct the logical circuit
   $F(a,b,c,d) = \Sigma(1, 2, 4, 5, 6, 7,8, 11, 12, 14) + \Sigma\Phi(3, 10, 13)$
5. Do the following conversions
   a. $(934.89)_{10}$ to Binary
   b. $(101.111)_2$ to decimal
   c. $(939AB)_{16}$ to decimal
   d. $(86.37)_{10}$ into octal
   e. $(76.35)_8$ into binary

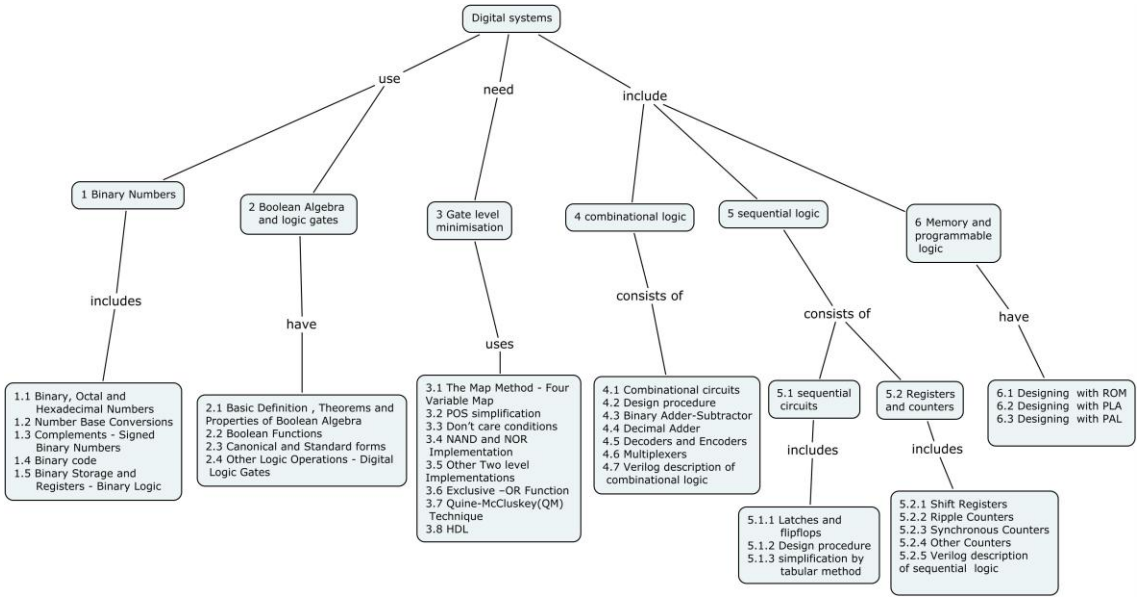
Create

1. Design a combinational circuit that converts a four bit BCD code to excess three code by using gates.
2. Design a synchronous counter using JK flip flop to generate the following Sequences.
   0,1,2,4,5,6.
3. Design a four bit binary synchronous counter with D flip- flops.
4. Produce the following systems to one with minimum number of states using the tabular method.

	Q^*		
q	X=0	x=1	z
A	C	B	0
B	D	A	1
C	A	B	0
D	B	B	1

5. Design a modulo-16 counter.

Concept Map



Syllabus

Digital Systems: - Binary numbers, Octal and Hexadecimal Numbers- Number Base Conversions – Complements - Signed Binary Numbers - Binary Codes - Binary Storage and Registers - Binary Logic - **Boolean Algebra and Logic Gates:** Boolean Algebra - Basic definition -Theorems and Properties of Boolean Algebra - Boolean Functions - Canonical and Standard forms- other Logic Operations - Digital Logic Gates - **Gate Level Minimization:** The Map Method - Four Variable Map - Product of Sums(POS) simplification - Don't care conditions - AND and NOR Implementation - Other Two level Implementations - Exclusive–OR Function - Quine-McCluskey(QM) Technique **-**HDL**. Combinational Logic:** Combinational Circuits - Design Procedure - Binary Adder- Subtractor - Decimal Adder - Decoders and Encoders – Multiplexers - Verilog description of combinational logic .**Sequential Logic:** Sequential Circuits - Latches- Flip-flops - Design Procedure - Registers and Counters - Registers - Shift

Registers - Ripple Counters - Synchronous Counters - Other Counters - Verilog description of sequential logic .**Memory and Programmable Logic:** Designing with ROM- Designing with Programmable Logic Array - Designing with Programmable Array Logic.

References:

1. M. Morris Mano and Michael D.Ciletti, Digital Design, PHI, Fourth edition, 2008
2. Alan B. Marcovitz,Introduction to Logic Design. TMH edition, 2002.
3. Floyd, Digital fundamentals, Nineth Edition, Pearson Education India, 2005.

Course content and Lecture Schedule

No.	Topic	No. of Lectures
1.	**Digital Systems**	
1.1	Binary Numbers - Binary, Octal and Hexadecimal Numbers	1
1.2	Number Base Conversions	1
1.3	Complements - Signed Binary Numbers	1
1.4	Binary Codes	1
1.5	Binary Storage and Registers - Binary Logic	1
2.	**Boolean Algebra and Logic Gates**	
2.1	Basic Definition , Theorems and Properties of Boolean Algebra	1
2.2	Boolean Functions	1
2.3	Canonical and Standard forms	1
2.4	Other Logic Operations - Digital Logic Gates	1
3.	**Gate Level Minimization**	
3.1	The Map Method - Four Variable Map	2
3.2	Product of Sums(POS) simplification	1
3.3	Don’t care conditions	1
3.4	NAND and NOR Implementation	1

3.5	Other Two level Implementations	1
3.6	Exclusive –OR Function	1
3.7	Quine-McCluskey(QM) Technique	2
3.8	HDL	1
4	**Combinational Logic**	
4.1	Combinational Circuits	1
4.2	Design Procedure	1
4.3	Binary Adder-Subtractor	2
4.4	Decimal Adder	1
4.5	Decoders and Encoders	2
4.6	Multiplexers	1
4.7	Verilog description of combinational logic	2
5.	**Sequential Logic**	
5.1	Sequential Circuits	1
5.1.1	Latches - Flip-flops	1
5.1.2	Design Procedure	2
5.1.3	Simplification by tabular method for state reduction	2
5.2	**Registers and Counters**	
5.2.1	Registers	1
5.2.2	Shift Registers	2
5.2.3	Ripple Counters	1
5.2.4	Synchronous Counters	1
5.2.5	Other Counters	2
5.2.6	Verilog description of sequential logic	2
6.	**Memory and Programmable Logic**	
6.1	Designing with Read-only Memory	1

6.2	Designing with Programmable Logic Array	2
6.3	Designing with Programmable Array Logic	2
	Total	**49**

Course Designer:

1. Mr. R.Chellamani rcmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 23	3	-	-	3

C23 Computer Organization 3:0

Preamble: This is a course offered in second semester for the students of Computer Science and Engineering. This course is preceded by the course "Basics of Electrical and Electronic Engineering" in the first semester to students of all branches. It sensitizes the students to issues of processing signals in digital and analog forms, beyond which students have no exposure to digital logic or function design of digital systems. This is the only course that is concerned with what a computer is at logic and organization level. There is no other course either in digital systems or in computer architecture and design offered to the students of Computer Science and Engineering. It aims to make the student familiar with basic computer logic and functional design of arithmetic and logic unit that is capable of performing floating point arithmetic operations. Different processor architectures are needed to meet different performance requirements. Advanced Processor architecture design and the performance analysis make the students to understand the real time scenario of internal system organization.

Program outcomes addressed

a. An ability to apply knowledge of engineering, information technology, mathematics, and science

c. An ability to design a system or component, or process to meet stated specifications

f. An ability to communicate effectively in English in both oral and written forms

h. An ability to engage in life-long learning

Competencies

1. Understand the Design of Internal features of computer.
2. Analyze the Performance of the system using Amdahl's Law.
3. Design a simple computer at functional level for a given set of specifications.
4. Design an instruction set of a simple computer capable of performing a specified set of operations.
5. Design the structure of registers of a CPU capable of executing a set of specified instructions.

6. Design a memory system for a given set of specifications.
7. Specify architectures using pipelining and paralleling features to improve the performance of computers.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
1	Remember	10	10	10	10
2	Understand	30	10	20	20
3	Apply	40	40	40	40
4	Analyze	10	10	10	10
5	Evaluate	0	0	0	0
6	Create	10	20	20	20

Course Level Learning Objectives

Remember

1. Define Accumulator features
2. Describe the format for Memory Reference Instruction?
3. State the sequence of Instruction Cycle?
4. Define Memory Function Completed
5. State Word line and Bit line in Memory
6. Identify the Multiple Bus benefits
7. Is Pentium processor a RISC or CISC processor?
8. List the features of Cache memory
9. Arrange the memories with respect to their speed and cost
10. Define Amdahl's Law
11. Describe Structural Hazard

Understand

1. Compare Micro programmed and Hardwired Control Unit?
2. Describe Complete Instruction
3. Illustrate the Complete Processor with diagram
4. Discuss the features of CISC
5. Compare RAM and ROM

6. Describe the print routine and computer routine handling by interrupt
7. Summarize the vector processor features
8. Compare PCI and SCSCI

Apply

1. Illustrate the Three Instruction Code formats
2. Identify the difference in BUN and BSA memory reference instructions
3. Demonstrate the control unit organization in case of conditional branching in Micro programmed control unit
4. Construct the Hardwired Control Unit with diagram
5. Illustrate the connection and control signals for register MDR
6. Demonstrate the 128x8 memory chip organization and how read/write is performed
7. Construct the Printer to Processor interface connection and explain its operation
8. Classify the hazards in Pipelining
9. Construct the Array Processor and its features

Analyze

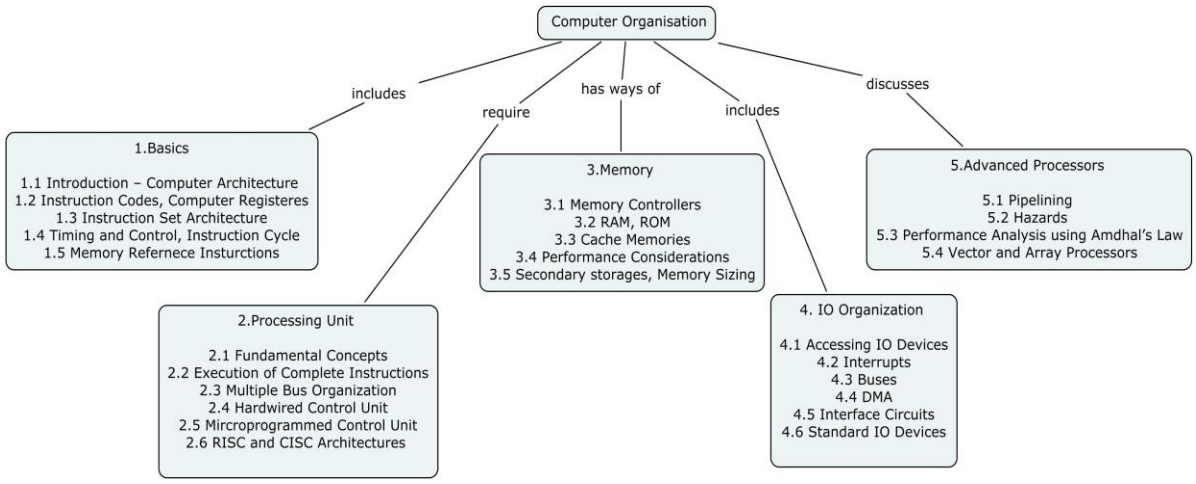
1. Compare Memory Reference and Register Reference Instructions
2. Predict the role of sequence counter used inside the control unit
3. Figure the partial format for field encoded microinstruction
4. Rate the Memory Hierarchy with respect to speed and cost
5. Compare associative and set associative cache memory
6. Sketch the Deisy Chain interrupt and priority group Interrupt schemes
7. Figure the open driven bus used to implement a common interrupt request line
8. Sketch the different interface standards used inside the system
9. Find the ways to avoid structural hazard

Create

1. Construct the role or Program Counter within CPU
2. Formulate CMOS memory cell storing concept with diagram
3. Devise the internal organization of 2M X 8 Dynamic Memory Chip
4. Design the ROM cell
5. Formulate the exception handling methods
6. Design the Bus Arbitration used in DMA

7. Formulate the interface circuit lines with diagram and discuss the features
8. Prepare the ways to measures the performance of the system using Amdahl's Law

Concept Map



Syllabus

Basic Computer Organization - Introduction – Computer Architecture, Instruction Codes, Computer Registers, Instruction Set Architecture, Timing and Control, Instruction Cycle, Memory Reference Instructions; **Processing Unit –** Fundamental Concepts, Execution of Complete Instructions, Multiple Bus Organization, Hardwired Control Unit, Micro programmed Control Unit, RISC and CISC Architectures; **Memory –** Memory Controllers, RAM, ROM, Cache Memories, Performance Considerations, Secondary storages, Memory Sizing; **IO Organization –** Accessing IO Devices, Interrupts, Buses, DMA, Interface Circuits, Standard IO Devices; **Advanced Processors** – Pipelining, Hazards, Performance Analysis using Amdahl's Law, Vector and Array Processors.

References:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, " Computer Organization" Tata Mc Graw Hill, Fifth Edition
2. M.Morris Mano, "Computer System Architecture" Pearson Third Edition

3. John L . Hennessy, David Patterson, " Computer Architecture- A Quantitative Approach" Elsevier Fourth Edition

Course Content and Lecture Schedule

No.	Topic	No. of Lectures
1	**Basic Computer Organization**	
1.1	Introduction – Computer Architecture	2
1.2	Instruction Codes, Computer Registers	1
1.3	Instruction Set Architecture	2
1.4	Timing and Control, Instruction Cycle,	2
1.5	Memory Reference Instructions	1
2	**Processing Unit**	
2.1	Fundamental Concepts	1
2.2	Execution of Complete Instructions	1
2.3	Multiple Bus Organization	1
2.4	Hardwired Control Unit	2
2.5	Micro programmed Control Unit	2
2.6	RISC and CISC Architectures,	1
3	**Memory**	
3.1	Memory Controllers	2
3.2	RAM,ROM	1
3.3	Cache Memories	2
3.4	Performance Considerations	1
3.5	Secondary storages, Memory Sizing	2
4.	**Input / Output Organization**	

4.1	Accessing IO Devices	1
4.2	Interrupts	2
4.3	Buses	1
4.4	DMA	1
4.5	Interface Circuits	1
4.6	Standard IO Devices	1
5	**Advanced Processor**	
5.1	Pipelining	2
5.2	Hazards	2
5.3	Analysis using Amdahl's Law	2
5.4	Vector and Array Processors	2

Course Designer:

1. **Ms.D.Tamilselvi dtamilselvi@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 24	3	-	-	3

C24 Procedural Programming 3:0

Preamble: The course on Procedural programming is intended to introduce the students to computational thinking, procedural programming and constructs of C programming. On Completion of the course students would be able to master structured programming and demonstrate applications on top of Procedural programs.

Program Outcomes addressed

a. An ability to apply knowledge of engineering, information technology, mathematics, and science

c. An ability to design a system or component, or process to meet stated specifications

Competencies

1. Select computers for different applications.
2. Comprehend the nature of problems that a computer can solve extremely well - be able to list 5 non-trivial, interesting problems (unique in their own way) which are difficult to solve for a human being but can be solved easily by a computer.
3. Comprehend the following terms in the context of problem solving by a computer: Problem specification, input-output analysis, algorithm, flowchart, pseudo-program, programming language, assembly language, machine language, compiler, assembler, program correctness
4. Explain the difference between arrays and linked lists, and create two examples where arrays are better than linked lists and two examples where linked lists are better than arrays.
5. Explain the difference between iteration and recursion, and create two examples where iteration is better than recursion and two examples where recursion is better than iteration.
6. Design the flowchart and write efficient code for problems like
   - Recursive and iterative programs for binary search
   - Recursive and iterative programs for Fibonacci numbers
   - Recursive and iterative programs for finding the GCD of two numbers
   - Reverse a linked list while traversing it only once

7. Explain the role of pointers in implementing singly linked lists, doubly linked lists, binary trees, and general trees.
8. Explain the reason why different constructs are available for iteration, such as "for" loops, "do...while" loops.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
1	Remember	20	10	10	0
2	Understand	20	20	20	10
3	Apply	50	40	40	50
4	Analyze	10	20	20	20
5	Evaluate	0	10	10	20
6	Create	0	0	0	0

Course Level Learning Objectives:

Remember

1 Define a Computer.
2 Mention the different I/O devices used with a computer.
4. List five programming languages commonly used
5. Define structured programming.
6. List out the various data types.
7. List down the various I/O functions in C.
8. Define format specifiers.
9. Define a header file.
10. List out the various looping statements.
11. Give the Syntax of switch-case statement.
12. Define an array.
13. Define a pointer.

Understand:

1. Compare while loop with do – while Loop.
2. Explain how recursive functions affect the run time efficiency.
3. Differentiate between Structure and Union in C.
4. Explain how dynamic arrays are efficient compared to Static with example.
5. Explain memory management in C.

6. List out the advantages of using Command line Arguments.
7. Explain garbage collection in C.

Apply

1. Write a recursive function to calculate the Combinatory of a nCr
2. Write a program to perform stack operation using pointers
3. Write a program to perform linked list operation using pointers
4. Write a program to generate the pay slip of an employee using dangling if – else statement.
5. Write a program to compute Matrix Multiplication using Pointers
6. Write a program to check if the given word is available in the file or not.

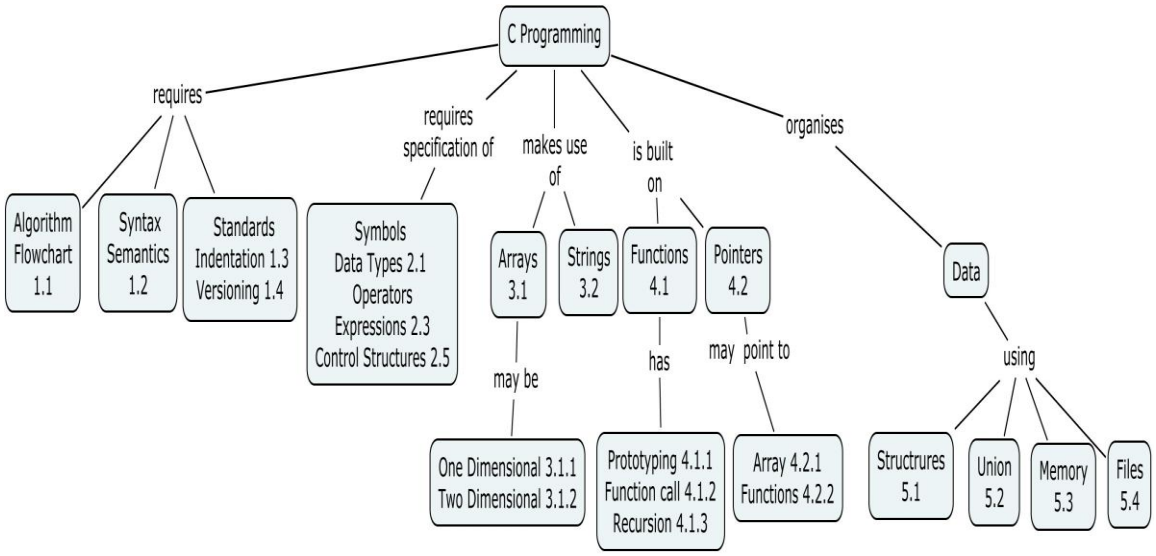
Analysis

1. Explain the difference between the following:
   (i) Program testing and debugging.
   (ii) Top down and bottom up approaches.
   (iii) Interpreted and compiled languages
2. Justify pointers to be powerful? Analyze its Efficiency giving an example.
3. Are the following statements valid? Justify your answer
   (i) k = (char*)& m
   (ii) m= (float*)& p
4. Is there any advantage of using recursion over looping control structures? Give a suitable example.
5. Analyze the factors that influence the execution times of a program.
6. Diagrammatically represent the functional blocks of a 1 pass assembler.
7. Outline the functions of the various loader schemes.
8. Illustrate the **Limitation of array of pointers to strings** using a sample.
9. Differentiate the keywords **BREAK** and **CONTINUE.**

Evaluate

1. Execution time affects efficiency more than storage space .Justify.
2. Justify the need for **Type Casting** over **Type Conversion**.
3. Compare and contrast **IO mapped IO** with **Memory mapped IO.**
4. Summarize the various built in **String** functions
5. Evaluate the sorting procedure using arrays and pointers.
6. Given an Educational institution try to automate it acquiring the needed resources.

Concept Map:



Syllabus:

Introduction to computers: Application Programming: Algorithms, Flowcharts, Syntax, semantics and execution, **Structured Programming Language:** Symbols and data types, Operators and Expressions, Storage classes, Looping control structures, Decision control structures, Case control structures. **Arrays and Strings** One dimensional array Two dimensional array, Introduction to Strings, Standard Library Functions, String manipulation applications **Functions and Pointers:** Functions prototyping, Types of Functions, Function Call,recursion, pointers, pointers to arrays, pointers to functions. **Data Handling**: Structures, Union, Memory allocation, File and I/0.

References:

1. Yashavant Kanetkar: Programming in ANSI C, 2nd Edition-BPB Publications
2. Byron Gottfried, "Programming with C", 2nd Edition, (Indian Adapted Edition), TMH publications, 2006.
3. Yeshavant Kanetkar: Understanding Pointers in C, 2nd Edition BPB Publications
4. Yashavant Kanetkar: Let us C, BPB Publications 8th Edition 2007
5. Behrouz A. Forouzan and Richard. F. Gilberg, "A Structured Programming Approach Using C", II Edition, Brooks–Cole Thomson Learning Publications,(2007).

Course content and Lecture Schedule

No.	Topic	No. of Lectures
1	**Applications Programming (6)**	
1.1	Algorithms and Flowcharts	1
1.2	Syntax, semantics and execution	1
1.3	Coding Standards and Indentation	2
1.4	Version control and maintenance	2
2	**Structured Programming Language (8)**	
2.1	Symbols and data types	1
2.2	Operators and Expressions	2
2.3	Storage classes	1
2.4	Looping control structures	2
2.5	Decision control structures	1
2.6	Case control structures	1
3.	**Arrays and Strings (7)**	
3.1.1	One dimensional array	2
3.1.2	Two dimensional array	2
3.2.1	Introduction to Strings	1
3.2.2	Standard Library Functions	1
3.2.3	String manipulation applications	1
4.	**Functions and Pointers (11)**	
4.1.1	Functions Protyping	2
4.1.1a	Types of Functions	1
4.1.2	Function Call by value , Call by reference	2
4.1.3	Recursion	1
4.2	Introduction to pointers	1

No.	Topic	No. of Lectures
4.2.1	Pointers to array	2
4.2.1	Pointers to function	2
5	**Data Handling(8)**	
5.1	Structures	2
5.1.1	Array of Structures	2
5.2	Union	1
5.3	Memory allocation	1
5.4	File and I/O	2
	Total Hours	**40**

Course Designer:

1. Mr. S. Prasanna sprcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 25	3	0	-	3

C25 Data Structures 3:0

Program Outcomes addressed

a. An ability to apply knowledge of engineering, information technology, mathematics, and science

c. An ability to design a system or component, or process to meet stated specifications

d. An ability to identify, formulate and solve engineering problems

Competencies

1. Ability to identify and implement appropriate data structure for a given application
2. Comprehend the terms "data abstraction", "abstract data type", and "data structures", and how data structures and algorithms have to be blended carefully to obtain efficient implementations.
3. Explain the notion of time complexity and the asymptotic notions of "Big Oh" with non-trivial examples.
4. Explain the difference between worst case complexity and best case complexity. Justify with an example algorithm for each of the complexities: O(n), O(n*2), O(n*3), O(2**n), O(n log n), O(n*2 log n),O(log n), O(log log n), O(sqrt(n)).
5. Identify all the trade-offs involved in choosing static versus dynamic data structures
6. In the context of searching, identify the trade-offs involved in selecting the most efficient data structure.
7. In the context of sorting, identify the trade-offs involved in selecting: (a) bubble-sort (b) insertion sort (c) selection sort (d) quick sort (e) merge sort (f) heap sort.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	20	10
2	Understand	30	20	10
3	Apply	20	30	30
4	Analyze	10	20	20
5	Evaluate	10	10	30
6	Create	0	0	0

Course Level Learning Objectives:

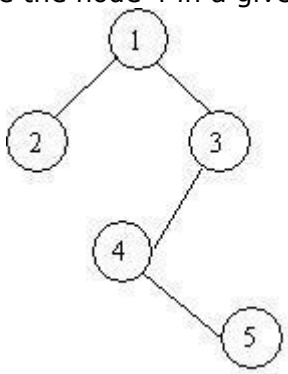
Remember

1. What is data structure?
2. List out the areas in which data structures are applied extensively?
3. What are the major data structures used in the following areas: RDBMS, Network data model and Hierarchical data model?
4. What are the notations used in Evaluation of Arithmetic Expressions using prefix and postfix forms?
5. List out few of the applications of tree data-structure?
6. List out few of the applications that make use of Multilinked Structures?
7. What is the bucket size, when the overlapping and collision occur at same time?
8. What are the Collision Resolution Techniques and the methods used in each of the type?
9. Draw a hash table with open addressing and a size of 9. Use the hash function "k%9". Insert the keys: 5, 29, 20, 0, 27 and 18 into your table (in that order).
10. Suppose that an open-address hash table has a capacity of 811 and it contains 81 elements. What is the table's load factor? (An appoximation is fine.)

Understand

1. If you are using C language to implement the heterogeneous linked list, what pointer type will you use?
2. What is the minimum number of queues needed to implement the priority queue?
3. How many null branches are there in a binary tree with 20 nodes?
4. How many different trees are possible with 10 nodes?

5. What is the condition for balancing to be done in an AVL tree?
6. How do you traverse a given tree using Inorder, Preorder and Postorder traversals.
7. What is the suitable efficient data structure for constructing a tree?
8. There are 8, 15, 13, 14 nodes were there in 4 different trees. Which of them could have formed a full binary tree?
9. At what location can you store the node 4 in a given binary tree using array?



10. Sort the given values using Quick Sort?

65	70	75	80	85	60	55	50	45

11. Classify the Hashing Functions based on the methods by which the key value is found.
12. What are the steps to inserting a new item at the head of a linked list? Use one short English sentence for each step.
13. Suppose that p is a reference to an IntNode in a linked list, and it is not the tail node. What are the steps to removing the node after p? Use one short English sentence for each step.
14. Write a class definition that could be used to define a node in a doubly linked list. Include only the instance variables, not the methods. Also write one sentence to describe a situation when a doubly linked list is appropriate.
15. Describe a situation where storing items in an array is clearly better than storing items on a linked list.
16. Describe why it is a bad idea to implement a linked list version a queue which uses the head of the list as the rear of the queue.

Apply

1. Convert the expression ((A + B) * C - (D - E) ^ (F + G)) to equivalent Prefix and Postfix notations.
2. Draw the B-tree of order 3 created by inserting the following data arriving in sequence - 92 24 6 7 11 8 22 4 5 16 19 20 78
3. Draw a binary Tree for the expression : A * B - (C + D) * (P / Q)
4. Is a Linked List a linear or non-linear data structure?
5. Suppose we are using the usual IntNode class (with instance variables called data and link). Your program is using an IntNode variable called head to refer to the first node of a linked list (or head is null for the empty list). Write a few lines of C++ code that will print all the double numbers on the list?
6. Suppose we are using the usual IntNode class (with instance variables called data and link), and that locate is referring to a node in a linked list. Write an assignment statement that will make locate refer to the next node in the list (if there is one). If there is no next node, then your assignment statement should set locate to null.
7. Suppose that p, q, and r are all references to nodes in a linked list with 15 nodes. The variable p refers to the first node, q refers to the 8th node, and r refers to the last node. Write a few lines of code that will make a new copy of the list. Your code should set THREE new variables called x, y, and z so that: x refers to the first node of the copy, y refers to the 8th node of the copy, and z refers to the last node of the copy. Your code may NOT contain any loops, but it can use the other IntNode methods.

Analyze

1. Why is the order of an algorithm generally more important than the speed of the processor?
2. Convert each time formula to the best possible big-O notation. Do not include any spurious constants in your big-O answer.

Time Formula	**Big-O**
10n	.
$2n^2$	.
3 times log (base 2) of n	.
$2n^2 + 10n$	.

3. Which of these is the correct big-O expression for 1+2+3+...+n?

   A. O(log n)
   B. O(n)
   C. O(n log n)
   D. O(n^2)

4. Which of the following formulas in big-O notation best represent the expression $n^2+35n+6$?

   A. O(n^3)
   B. O(n^2)
   C. O(n)
   D. O(42)

5. Answer true or false for this statement: For all possible inputs, a linear algorithm to solve a problem must perform faster than a quadratic algorithm to solve the same problem.

   - TRUE
   - FALSE

6. Answer true or false for this statement: True or false: An algorithm with worst case time behavior of 3n takes at least 30 operations for every input of size n=10.

   - TRUE
   - FALSE

7. What term is used to describe an O(n) algorithm.

   A. Constant
   B. Linear
   C. Logarithmic
   D. Quadratic

8. Here is some code for an *integer* variable n:

```
while (n > 0)
{
   n = n/10; // Use integer division
}
```

   What is the worst-case time analysis for the above loop?

   A. O(1)
   B. O(log n)
   C. O(n)
   D. O(n^2)

9. Express the formula (n - 2)*(n - 4) using big-O notation:

   A. O(1)
   B. O(8)
   C. O(log n)
   D. O(n)
   E. None of the above

10. Fill in the following table for the times to sort an array of n items. Use only big-O notation, and do not have any extraneous constants in your expressions.

	Worst Case	**Average Case**
Binary search of a sorted array	.	.
Insertion sort	.	.
Merge sort	.	.
Quick sort without "median of three" pivot selection	.	.
Quick sort with "median of three" pivot selection	.	.
Selection sort	.	.
Heap sort	.	.

Evaluate

1. Compare the worst-case big-O time analysis for these two methods: The add method for the Bag that is implemented using an array, and the add method for the Bag that is implemented using a linked list.
2. Compare the worst-case big-O time analysis for these two methods: The remove method for the Bag that is implemented using a fixed-sized array, and the remove method for the Bag that is implemented using a linked list.
3. Compare the worst-case big-O time analysis for these two methods: The addBefore method for the Sequence that is implemented using an array, and the addBefore method for the Sequence that is implemented using a linked list.
4. Compare the worst-case big-O time analysis for these two methods: The remove method for the Sequence that is implemented using an array, and the remove method for the Sequence that is implemented using a linked list.
5. I am going to execute this code with THREE pushes and ONE pop:

```
IntStack s = new IntStack( );
s.push(1);
s.push(2);
s.push(3);
System.out.println(s.pop( ));
```

Suppose that s is represented by a linked list. Draw the state of the private member variables of s after the above code:

6. Implement the following method. You may use the IntStack class and the Stack operations of push, pop, peek, isEmpty, and size. The parameter, in, is an EasyReader from Appendix B of the text and it is already attached to some kind of input. You may use the methods:

   in.isEOLN() -- returns true when the end of line is reached.
   in.peek() -- returns the next input character without actually reading it.
   in.ignore() -- reads and throws away the next input character.
   in.intInput() -- reads and returns an integer value from the EasyReader.

This should be used only if you know that the next input characters form a valid integer value.

The method specification is:

   public static int evaluatePostfix(EasyReader in)
   Precondition (Which is not checked): The next input line of in is a
   properly formed postfix expression consisting of integers,
   the binary operations + and -, and spaces.
   Postcondition: The method has read the next input line (including
   the newline) and returned the value of the postfix expression.

7. Consider the usual algorithm to convert an infix expression to a postfix expression. Suppose that you have read 10 input characters during a conversion and that the stack now contains these symbols: +(top), (, *(bottom)Now, suppose that you read and process the 11th symbol of the input. Draw the stack for the case where the 11th symbol is:

   A. A number:

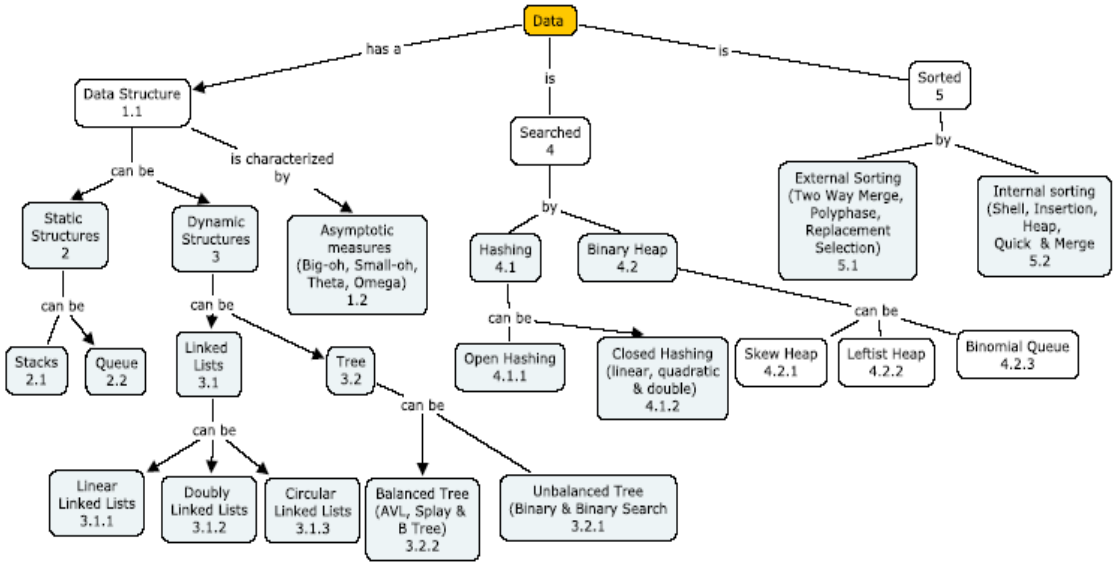
   B. A left parenthesis:

   C. A right parenthesis:

   D. A minus sign:

   E. A division sign:

Concept Map



Course Contents:

1. Data

1.1 Data Structure

1.2 Asymptotic Measures

2. Static Data Structures

2.1 Stacks

2.2 Queues

3. Dynamic Data Structures

3.1 Linked Lists

- 3.1.1 Linear Linked Lists
- 3.1.2 Doubly Linked Lists
- 3.1.3 Circular Linked Lists

3.2 Trees

- 3.2.1 Unbalanced Trees
- 3.2.2 Balanced Trees

4. Data Search

4.1 Hashing

- 4.1.1 Open Hashing
- 4.1.2 Closed Hashing

4.2 Heap

- 4.2.1 Skew Heap

4.2.2 Leftist Heap

4.2.3 Binomial Queue

5 Data Sorting

5.1 Internal Sorting

5.2 External Sorting

Syllabus

Data: Data Structure, Asymptotic Measures **Static Data Structures:** Stacks, Queues **Dynamic Data Structures:** Linked Lists: Linear Linked Lists, Doubly Linked Lists and Circular Linked Lists, Trees: Unbalanced and Balanced Trees, **Data Search:** Hashing: Open Hashing and Closed Hashing; Heap: Skew Heap, Leftist Heap, Binomial Queue **Data Sorting:** Internal Sorting: Insertion sorting, Shell sorting, Quick sorting, Merge sorting and Heap sorting; External Sorting

Textbook

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

Reference

1. Mark Allen Weiss: Data Structures and Algorithms in C, Addison-Wesley, 1997

Lecture Schedule

No.	Topic	No. of Lectures
1	**Data**	
1.1	Data Structure	0.5
1.2	Asymptotic Measures	1
2	**Static Data Structures**	
2.1	Stacks	3
2.2	Queues	2
3	**Dynamic Data Structures**	
3.1	Linked Lists	0.5
3.1.1	Linear Linked Lists	2
3.1.2	Doubly Linked Lists	1.5
3.1.3	Circular Linked Lists	1.5

No.	Topic	No. of Lectures
3.2	Trees	1
3.2.1	Unbalanced Trees	3
3.2.2	Balanced Trees	6
4	**Data Search**	
4.1	Hashing	0.5
4.1.1	Open Hashing	1
4.1.2	Closed Hashing	2.5
4.2	Heap	2
4.2.1	Skew Heap	1.5
4.2.2	Leftist Heap	1.5
4.2.3	Binomial Queue	1
5	**Data Sorting**	
5.1	Internal Sorting	
5.1.1	Insertion sorting	1
5.1.2	Shell sorting	1
5.1.3	Quick sorting	1
5.1.4	Merge sorting	1
5.1.5	Heap sorting	2
5.2	External Sorting	2

Course Designers:

1. **M.K. Kavitha Devi** **mkkdit@tce.edu**
2. **S. Sudha** **ssj@tce.edu**
3. **S. Geetha** **Sgeetha@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 26	2	-	-	2

C26 Environment and Ecology 2:0

(Common to all branches of Engineering B46, D26, E26, G36, T26)

Preamble: Environmental science is a subject carries too much of importance in the modern world.The basic knowledge on environment is an absolute necessity for all citizens.This course work aims at imparting the fundamental knowledge on environmental science which includes natural reasources and their exploitation, Ecosystem, Biodiversity and its conversation Envronmental pollution.

Programme outcomes addressed

- Graduates will demonstrate understanding of the societal, health, safety, legal and cultural issues and consequent responsibilities relevant to engineering practice.
- Graduates will understand the impact of engineering solutions in the societal contest and demonstrate knowledge of and need for sustainable development.
- Graduate will understand and commit to professional ethics and responsibilities and norms of engineering practice.

Competencies:

At the end of the course the student should be able to

(i) Understand the significance of conversation of natural reasources.

(ii) Realize that who is also one of the components of the ecosystem like any other species.

(iii) Appreciate the values of biodiversity and necessity of its conversation.

(iv) Find out the various sources, effects and control measures of environmental pollution.

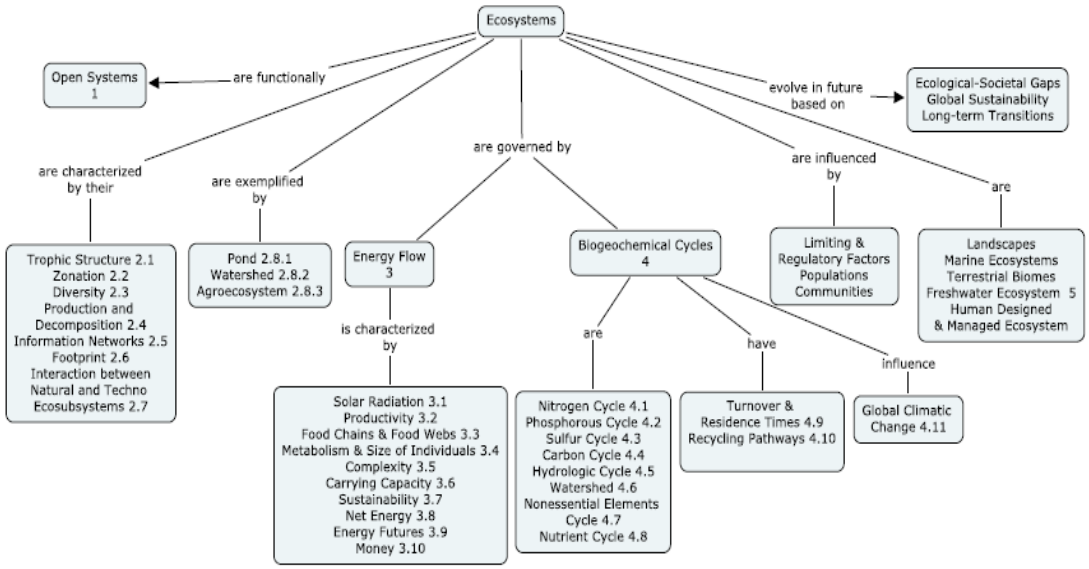
(v) Deliver the best individually towards environment sustainability.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	20	20
2	Understand	40	40	40
3	Apply	30	40	40

4	Analyze	-	-	-
5	Evaluate	-	-	-
6	Create	-	-	-

Concept Map



Course contents and Lecture Schedule

S.No.	Topics	Periods
1	**Natural Reasources**	
1.0	Multidisciplinary nature of Environmental studies & public awareness	1
1.1	Forest – uses, over exploitation, deforestation	1
1.2	Water – uses, over utilization, floods,droughts	**1**
1.3	Food – agriculture overgrazing, modern agriculture. Fertilizer – pesticide problems , water logging and salinity	1
1.4	Energy – renewable , non renewable energy, use of alternate energy	1
1.5	Land – degradation, land slides, soil erosion, desertification.	1
2	**Ecosystem**	1

S.No.	Topics	Periods
2.1	Ecosystem – Concept , structure, function, components	1
2.2	Laws of Ecology	1
2.3	Energy flow in ecosystem – Food chains,food webs	1
2.4	Ecological pyramids – Ecological succession	1
2.5	Types of ecosystem – Forest grass land, desert, aquatic ecosystem	1
2.6	Loss of Ecosystem and its Estimation	1
3	**Biodiversity and its conservation**	
3.1	Types of Biodiversity, biogeographical and classification	1
3.2	Values of Biodiversity	1
3.3	Hotspots of Biodiversity and Threats to Biodiversity	1
3.4	Biodiversity indices	1
3.5	Endangered and endemic species-conservation of biodiversity	1
4	**Environmental pollution and control**	
4.1	Causes ,effects and control measures of air pollution	1
4.2	Causes ,effects and control measures of water pollution	1
4.3	Causes ,effects and control measures of soil pollution	1
4.4	Causes ,effects and control measures of noise pollution	1
4.5	Solid Waste Management – Causes effects and control measures	1
4.6	Water conservation – Rainwater harvesting – Global Warming	1
4.6.1	Climate change and its effect on environment	1
4.7	Ozone layer depletion – legal provisions	1

Syllabus

General - Multidisciplinary nature of environmental studies –need for public awareness. Natural reasorces – Forest – uses, over exploitation deforestation and their effects. Water – uses, over utilization , floods , droughts. Food – architecture overgrazing, modern agriculture. Fertilizer – pesticide problems, water logging and salinity. Energy – growing needs , renewable , non renewable energy, use of alternate energy. Land – degradation land slides ,soil erosion ,desertification.

Ecosystem - Concept, structure , function, components. Laws of Ecology, Energy flow in ecesytem. –Food chains, food webs Ecological pyramids – Ecological sucession. Types of ecosystem – Forest grass land,desert,aquatic ecosyatem, Loss of ecosystem and its Estimation.

Biodiversity and its conversation – biodiversity types, biogeographical classification. Values of biodiversity – Hot spots of biodiversity – threats to biodiversity – Biodiversity Indices – Endangered and endemic species – conservation of biodiversity.

Environmental pollution – Causes , effects and control measurements of air pollution – water pollution – soil pollution – Noise pollution. Solid waste management – causes effects and control measures.

Water conservation - Rainwater harvesting – global warming – Climate Change and its effect on Environment – ozone layer depletion – legal provisions.

Text Book

1. Kaushik and Kaushik, 'Environmental Science and Engineering', 2nd edition,New age International publishers, New Delhi, 2006

Reference Books

1. Wright and Nebel, 'Environmetal science towards a sustainable future', Prentice Hall of India Ltdd, 2000.
2. S.K. Garg and Garg, 'Ecologicl and Environmental studies', khanna publishers, Delhi, 2006.

3. Gillbert M.Masters, 'Introduction to Environmental Engineering and Science',Second edition, Pearson Education publication,Delhi,2004.

Course Designers

1. T. Velrajan tvciv@tce.edu
2. Mr.S.Chandran schandran@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 27	-	-	3	1

C27 Data Structures Laboratory 0:1

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.

List of Exercises:

1. Stack implementation using Array, Two way stack implementation
2. Expression evaluation using stack
3. Queue implementation using Array
4. Round Robin Scheduling using circular Queue
5. Implementation of basic operations like creation, insertion, deletion, finds and display in a Singly linked list.
6. Polynomial addition and subtraction using Singly Linked List
7. Josephus Problem using Circular List
8. Palindrome checking using doubly linked List
9. Implementation of basic operations like creation, insertion, deletion, finds and display in a Binary Search Tree.
10. Implementation of inorder, preorder and postorder tree traversal in Binary Search Tree.
11. Implementation of open hashing and closed hashing techniques for searching.
12. Design a suitable application for analyzing the performance of Insertion Sort, Merge Sort, Shell Sort, Quick Sort and Heap Sort Algorithms.

Course Designer:

1. Mrs. S. Sudha ssj@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 28	-	-	3	1

C28 Procedural Programming Laboratory 0:1

List of Exercises:

1. Simple Programs
    a. Fibonacci Series
    b. Sum of set of numbers
    c. Generation of prime numbers
2. Program demonstrating control structures.
3. Matrix Manipulation
4. Sorting of Names & Numbers
5. String Manipulation
6. Bitwise Operation
7. Recursive functions.
8. Pointers to functions
9. Array of Structures
10. Pointers to Pointers
11. File Manipulations
    a. Read the file and display the contents of the file
    b. Read the lines from the keyboard and write it into a specified file
12. Store and retrieve the structure elements in a specific file
13. Program for creating files with read and write permissions

Course Designer:

1. Mr. S. Prasanna sprcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 29	-	-	3	1

C29 Work Shop 0:1

(Common to all branches of Engineering B29, C29, D29, E29, G29, T29)

Objective: The students of all branches f engineering would get exposure to basic practices in a mechanical workshop. The students get trained to acquire skills at basic level in fitting, carpentry, joining, metal forming and plumbing.

List of Exercises

I Fitting (Any four exercises)

1. Fitting tools and practice
2. Joining of two different metals with adhesives
3. Preparation of single step joint
4. Preparation of 'V' joint
5. Preparation of Gauge joint
6. Preparation of Taper sep joint

II Carpentry (Any four exercises)

1. Carpentry tools and practice
2. Joining different types of wood with adhesives
3. Preparation of Half joint
4. Preparation of Dovetail joint
5. Preparation of T-brittle joint
6. Turning on wood lathe

III Demonstration on Tools and Practice (Any four exercises)

1. Welding
2. Soldering
3. Brazing
4. Foundry and Moulding practice
5. Smithy forging
6. Plumbing

Terminal Examination: Students are tested in fitting and carpentry trades

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

THIRD SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2000 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

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

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the year 2013-2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

Board of studies Meeting on 13.06.2009 Approved in 38th Academic Council meeting 27.06.2009

THIAGARAJAR COLLEGE OF ENGINEERING : MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014 onwards)

THIRD SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			Credits
			L	T	P	
THEORY						
C 31	Engineering Mathematics III	BS	4	-	-	4
C 32	Graphics and Visualization	DC	3	-	-	3
C 33	System software	DC	4	-	-	4
C 34	Operating systems: Configuration and Use	DC	3	-	-	3
C 35	Design and Analysis of Algorithms	DC	3	-	-	3
C 36	Object Oriented Programming	DC	3	-	1	4
PRACTICAL						
C 37	Operating systems and System Software Lab	DC	-	-	2	1
C 38	Object Oriented Programming Lab	DC	-	-	2	1
Total			**20**	**-**	**5**	**23**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
DC : Department Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS

(For the candidates admitted from 2013-2014 onwards)

THIRD SEMESTER

S.No.	Sub. code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C 31	Engineering Mathematics III	3	50	50	100	25	50
2	C 32	Graphics and Visualization	3	50	50	100	25	50
3	C 33	System software	3	50	50	100	25	50
4	C 34	Operating systems : Configuration and Use	3	50	50	100	25	50
5	C 35	Design and analysis of Algorithms	3	50	50	100	25	50
6	C 36	Object Oriented Programming	3	50	50	100	25	50
PRACTICAL								
7	C 37	Operating systems and System software Lab	3	50	50	100	25	50
8	C 38	Object Oriented Programming Lab	3	50	50	100	25	50

* Continuous Assessment evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub Code	Lectures	Tutorial	Practical	Credit
C 31	4	-	-	4

C31 Engineering Mathematics III 4:0

(Common to all branches of Engineering, B31,C31,D31,E31,G31,T31)

Preamble:

An engineering student needs to have some basic mathematical tools and techniques. 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 the adequate exposure in the theory and applications of Fourier series, Fourier Transforms, PDE's and BVP

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

j. Graduate will develop confidence for self education and ability for life-long learning.

Competencies

At the end of the course the student should be able to

1. Express the periodic functions arising in the study of engineering problems as Fourier series of Sines and Cosines.
2. Find the Fourier series for the typical waveforms.
3. Find the Fourier series for discrete data using Harmonic Analysis.
4. To study some of the well-known integral transforms (like Fourier, Fourier Sine and Cosine) and properties.
5. Formulate simple Engineering problems as Partial Differential Equations and state the boundary conditions.
6. Solve Partial Differential Equations, linear, nonlinear, homogeneous and non-homogeneous, by various methods.
7. Solve the standard Partial Differential Equations arising in engineering problems like Wave equation, Heat flow equation (one dimensional and two dimensional, Cartesian and polar coordinates) by Fourier series.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. Define Periodic function.
2. Show that f(x)= x^3 is an odd function.
3. State the Fourier Series for the function f(x) in the interval (1, 3).
4. Identify the Kernel for Fourier Cosine and Sine Transforms.
5. State Parseval's Identity.
6. State Convolution Theorem.

Understand

1. Distinguish between Odd and Even functions.
2. Use the Fourier series expansion of $x^2 = \frac{\pi^2}{3} + 4\sum_{1}^{\infty}(-1)^n \frac{\cos nx}{n^2}$, $-\pi < x < \pi$ to predict the value of $\sum \frac{1}{n^2}$.
3. Discuss harmonic analysis.
4. Discuss Fourier Series in Complex form.
5. Interpret the result $F[f(ax)] = \frac{1}{a} F\left(\frac{s}{a}\right)$.
6. Interpret the usage of Parseval's theorem.
7. Discuss the two methods of forming partial differential equations.
8. Discuss the solution of $\frac{\partial^2 z}{\partial x \partial y} = x^2 y$ by direct integration.
9. Discuss the working rule of solving the Lagrange's linear equation.
10. Discuss the working rule of solving f(p,q)=0.
11. Discuss the working rule of solving f(z,p,q)=0.

Apply

1. Find the Fourier transform of $e^{-a^2x^2}$. Hence prove that $e^{-\frac{x^2}{2}}$ is self reciprocal with respect to Fourier transforms and (i) Find the Fourier Cosine transform of e^{-x^2} .
2. Solve the equation $z^2(p^2+q^2+1)=c^2$ where c is a constant.
3. Obtain the first three harmonics in the Fourier series expansion in (0,12) for the function y=f(x) defined by the table given below:

X:	0	1	2	3	4	5	6	7	8	9	10	11
Y:	1.8	1.1	0.3	0.1	0.5	1.5	2.16	1.88	1.25	1.30	1.76	2

4. Find the Fourier transform of f(x), if $f(x)=\{1-|x|, for|x|<1 and 0, for|x|>1\}$.

Syllabus

Fourier Series: Dirichlet's conditions, General Fourier Series, Half range Sine and Cosine series, Parseval's Identity, Harmonic Analysis, Complex form of Fourier Series. **Fourier Transformation:** Fourier Integral Theorem, Fourier Transform, Fourier Sine and Cosine Transforms, Convolution Theorem, properties, Parseval's Identity, Discrete Fourier Transform, Discrete Time Fourier Transform, Demonstration of Fourier transforms and its properties using MATLAB (Tutorial).**Partial Differential Equations**: Formation, Solution of standard types of first order equations, Lagrange's linear equation, Linear partial differential equations of second and higher order with constant coefficient. **Boundary Value Problem**: Classification of Second Order linear partial differential equations, One-dimensional Wave equation, One dimensional heat equation, Solution by Fourier Series, Steady State Solution of two dimensional heat equation in Cartesian Co-ordinates, Laplace equation in Polar Co-ordinates, Solution by Fourier Series method.

Text Book

B.S. Grewal: Higher Engineering Mathematics, 39th Edn. , Khanna Publishers, New Delhi, 2007.

References

1. Lecture Notes by the faculty of Department of Mathematics, TCE ,Madurai.
2. Veerarajan .T: Engineering Mathematics, 3rd Edition. , Tata McGraw Hill, NewDelhi, 2004
3. Kreyszig,E., "Advanced Engineering Mathematics", John wiley and sons,(Asia)Pte Ltd., Singapore.2006.

Course contents and Lecture Schedule

No.	Topic	No. of Lectures

1.	**Fourier Series**	
1.1	Dirichlet's conditions, General Fourier Series	3
1.2	Half range Sine and Cosine series	3
1.3	Parseval's Identity	1
1.4	Harmonic Analysis	2
1.5	Complex form of Fourier Series	2
2.	**Fourier Transformation**	
2.1	Fourier Integral Theorem, Fourier Transform	2
2.2	Fourier Sine and Cosine Transforms	2
2.3	Convolution Theorem	1
2.4	Properties, Parseval's Identity	2
2.5	Discrete Fourier Transform, Discrete time Fourier Transform	2
2.6	Demonstration of Fourier transforms and its properties using MATLAB (Tutorial)	3
3	**Partial Differential Equations**	
3.1	Formation	2
3.2	Solution of standard types of first order equations	3
3.3	Lagrange's linear equation	2
3.4	Linear partial differential equations of second and higher order with constant coefficient	3
4	**Boundary Value Problems**	
4.1	Classification of Second Order linear partial differential equations	1

4.2	One-dimensional Wave equation, Solution by Fourier Series	4
4.3	One dimensional heat equation, Solution by Fourier Series	4
4.4	Steady State Solution of two dimensional heat equation in Cartesian Co-ordinates, Solution by Fourier Series	4
4.5	Laplace equation in Polar Co-ordinates, Solution by Fourier Series	4

Course Designers

1. V.Mohan vmohan@tce.edu
2. N. Kannan nkmat@tce.edu
3. N. Chitra ncmat@tce.edu
4. M.Sivanandha Saraswathy sivanandha@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 32	3	-	-	3

C32 Graphics and Visualization 3:0

Preamble: The goal of this subject is to motivate the students to acquire knowledge and skills in graphics and visualization. The topics like 2D, 3D objects and visualization help the students to implement their own ideas with more creativity.

Programme Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

f. Graduate will demonstrate skills to use modern engineering tools, softwares and equipment to analyze problems.

Competencies

1. Explain the role of graphics and images in human computer interface
2. Explain the functioning and characteristics of input and output devices
3. Explain and compare algorithms for line and circle drawings, clipping and transformations
4. Explain the differences between 2D and 3D graphics
5. Select 2D and 3D tools for different applications

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	10	10
2	Understand	20	30	20
3	Apply	20	30	40
4	Analyze	20	20	20
5	Evaluate	10	10	10
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. What is clipping? List out its types.
2. Name the basic two dimensional transformations.
3. Give the matrix representation for shearing.
4. What is composite transformation?
5. What is viewing and windowing?
6. Define projection.
7. Give the properties of B-Spline curve.
8. Define animation.
9. Mention some general purpose languages that support animation.

10. What is scientific visualization?
11. List out the visualization techniques.
12. What are the display methods available for representing 3D objects?
13. List out 2D and 3D tools.
14. What are the steps involved in designing animation sequence.
15. Give a procedure to design an object surface using Bezier curves?
16. Prove that the rotation matrix R is equal to the composite matrix R_y (ß). R_x(α).

Understand

1. Difference between raster and vector graphics.
2. What is the difference between 2D and 3D objects?
3. Illustrate the logic of the Sutherland –Hodgeman algorithm.
4. Explain the reflection and shear transformation in three dimensional graphics applications by providing the matrix representation and example figures.
5. Difference between window and view port.
6. How curves are used in modeling and function graphing?
7. Explain how straight lines can be used to draw Bezier cubic curves.
8. Describe the procedure for drawing Bezier curves.
9. How will you perform morphing? Mention its applications.
10. Explain how to create running tiger from moving automobile using morphing effect.
11. Write a procedure to create bouncing ball using motion specifications.
12. Differentiate parallel projection from perspective projection.

Apply

1. Perform a 45 deg. Rotation of triangle a (0, 0), b (1, 1), c (5, 2) about the origin and about p (1,-1).
2. Write a function to draw a rectangle.
3. Write a function to draw a polygon.
4. Formulate transformation matrices for the following: Rotation about any pivot point (xp, yp) not just origin, Scaling relative to any fixed point (xf, yf) not just origin.
5. Illustrate the mid point circle generation algorithm for drawing a circle centered at (0, 0) and having radius of 5 units.
6. Write a program to demonstrate the Reflection and Shear transformations.
7. Write the algorithm for point clipping.
8. Implement the Cohen-Sutherland Line Clipping Algorithm.
9. Write an efficient routine to display 2D cubic Bezier curve given a set of four control points in the xy plane.
10. Explain how visualization is used in different applications?.
11. Write a program to implement scan-line algorithm for displaying the visible- surfaces of a given polyhedron.

12. Write a program to perform text clipping.
13. Determine the sequence of basic transformations that are equivalent to the x-direction shearing matrix.
14. Determine the Bezier blending functions for five control points. Plot each function and label the maximum and minimum values.
15. Determine the blending functions for uniform, periodic B-Spline curves for d=5.

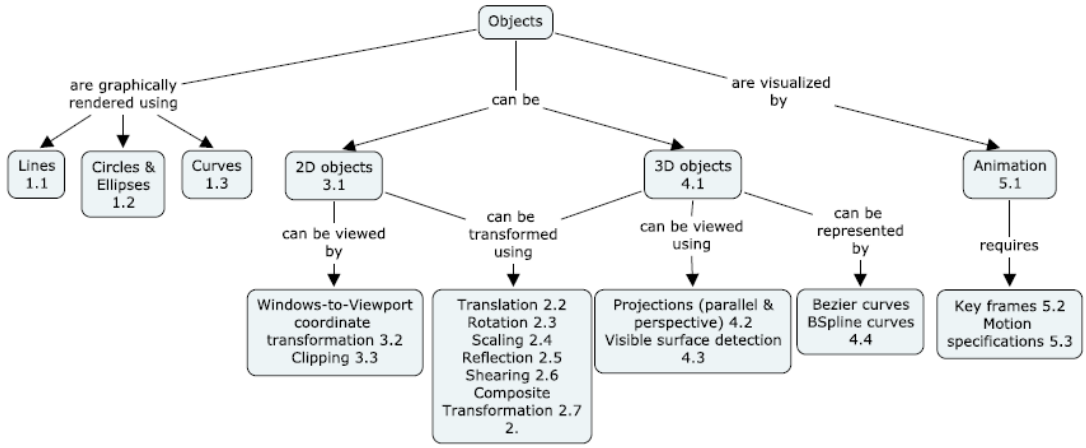
Analyze

1. Analyze the space and time complexity of two line drawing algorithms.
2. Analyze the computational efficiency in composite transformations.
3. Analyze the number of arithmetic operations performed in Cohen Sutherland and the Liang-Barsky line clipping algorithm for several different line orientations relative to clipping window.
4. Compare and contrast three methods of visible surface detection algorithm and hence analyze which algorithm is efficient? .
5. What are the advantages of B-Spline over Bezier Curves?

Evaluate

1. Evaluate that the multiplication of 2D transformation matrices for each of the following sequence of operation is commutative.
    a. Two successive Rotations.
    b. Two successive translations.
    c. Two successive scaling.

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. **2. 2D and 3D Geometric Transformations:** Basic Transformations- Translation, Rotation, Scaling, Reflection, Shearing, Composite Transformations **3. 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, Curve and Text Clipping. **4. 3D Viewing and Object Representation:** Viewing Pipeline, Projections – Parallel and Perspective Projection, View-port Clipping, Bezier Curves and Surfaces, B-Spline Curves and Surfaces, Visible Surface Detection Algorithm- Back-Face Detection Algorithm, Depth Buffer Method, Scan line Method. **5**. **Visualization and Animation:** Visualization and its techniques, Overview of Data Visualization, Applications of Visualization, Principles of animation, File Formats, Keyframes, Motion Specifications.

Text Book

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

References

1. Zhigand Xiang, Roy Plastock: Theory and problems of Computer Graphics, Schaum's outline Series, Tata Mc-Graw hill edition. 2005.
2. Foley, James D Dam, Andries Van: Computer Graphics Principles and Practice, Pearson Education, 2002.
3. 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
0	**Graphics and Visualization: an Introduction**	**1**
1.	**Rendering of Objects**	
1.1	Rendering of lines ((DDA Algorithm, Bresenham's Line Algorithm, and Parallel Line Algorithm)	3
1.2	Rendering of circles (Mid-point circle algorithm)	2
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	Introduction to Basic Transformations	1
2.2	Translation	1

2.3	Rotation	1
2.4	Scaling	1
2.5	Reflection	2
2.6	Shearing	2
2.7	Composite Transformations	2
3.	**2D Viewing**	
3.1	Viewing Pipeline	1
3.2	Window to view-port coordinate Transformation	1
3.3	Clipping	
3.3.1	Point Clipping	1
3.3.2	Line Clipping (Cohen Sutherland Line Clipping, Liang Barsky Line Clipping)	2
3.3.3	Sutherland Hodgeman Polygon Clipping	1
3.3.4	Curve and Text Clipping	1
4	**3D Viewing and Object Representation**	
4.1	Viewing Pipeline	1
4.2	Projections – Parallel and Perspective Projection	2
4.3	Visible surface detection (Back-Face Detection Algorithm, Depth Buffer Method, Scan line Method)	2
4.4	Bezier, B-spline curves and Surfaces	4
5	**Visualization and Animation**	
5.1	Visualization of objects	4
5.2	Key frames	1
5.3	Motion Specifications	1
	Total	**40**

Course Designers

1. S. Sridevi sridevi@tce.edu
2. R. Suganya rsuganya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 33	3	1	-	4

C33 System Software **4:0**

Preamble:

Prerequisite: C23 – Computer Organization and Design, C24 – Computers and Programming

To provide a basic knowledge in the concepts of system software and the architecture of various machines, to understand the working principle of assemblers, the functions of loaders, linkers and macro processors. At the end of the course, the students will be able to design an assembler, loader for a given architecture and they also know how to link libraries with high level programming languages.

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

j. Graduate will develop confidence for self education and ability for life-long learning.

Competencies

1. Explain the architectural features, including memory, registers, data format, instruction sets, instruction format and addressing modes of a given computer
2. Identify the features of assemblers, linkers, loaders, and macro processor for different computer architectures.
3. Design different elements of system software for a given architecture

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	20	10
2	Understand	50	60	60
3	Apply	20	20	30
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level Learning Objectives

Remember

1. Describe the addressing modes of SIC/XE
2. What is formal system?

3. Define assemblers, compilers.
4. What are program blocks?
5. Define control sections.
6. What is meant by bootstrap loader?
7. What are the data structures used in assemblers, loaders?
8. Define a macro.

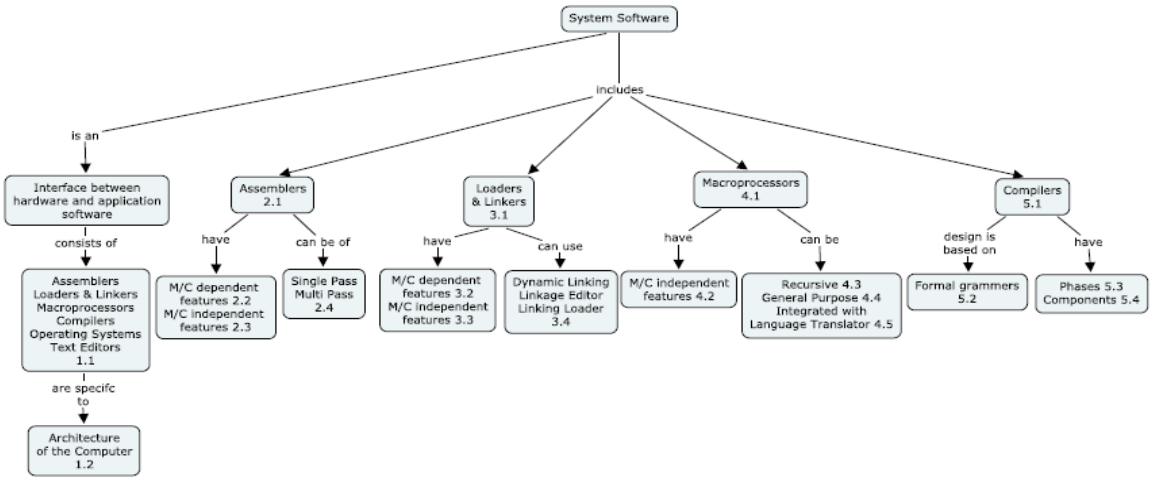
Understand

1. What is the purpose of a symbol table in assemblers?
2. How assembler directives are processed in assemblers?
3. What are the functions of an OS?
4. Write the difference between ORG and EQU.
5. What is meant by delayed binding? Mention its advantages.
6. Write the difference between subroutine and macro.
7. What is type1 and type 2 grammars? Give examples?
8. Why two pass assembler is needed?
9. How forward references are resolved in one pass assemblers.
10. How jump instructions are assembled in MASM assemblers.
11. Why loaders need 2 passes over a program?

Apply

1. Write a code in SIC/XE to read 200 byte record from the device 'F1" using subroutine?
2. Write a program in C for a single pass assembler for Pentium architecture?
3. Write a program in C for a multi pass assembler for Pentium architecture?
4. Write a program in C for a relocatable loader for Pentium architecture?
5. Write a program in C for a absolute loader for Pentium architecture?
6. Write a program in C for implementing a macroprocessor for Pentium architecture?

Concept Map



Syllabus

Introduction and Machine architecture - Assemblers, loaders, linkers, macros, compilers, operating system, formal system - Simplified instructional computers – architecture, programming. **Assemblers -** Basic functions, algorithm, data structures - Machine dependent features - Machine independent features - Single pass and Multi pass assemblers – Implementation Examples. **Loaders and Linkers -** Basic functions - Machine dependent features - Machine independent features - Dynamic Linking, linkage editor and linking loader – Implementation Examples. **Macro processors -** Basic functions - Machine independent features - Recursive macro processors - General purpose macro processors - Macro processors integrated with language translators – Implementation Examples. **Compilers -** Analysis of source program (formal grammar) - Phases of compiler - Components of compiler.

Text Books:

1. L.Beck, "System Software – An Introduction to Programming", Third Edition – Pearson Education Inc, 2004.
2. John J.Donovan, "System Programming", Tata McGraw Hill, 2000.
3. Alfred V. Aho, Jeffrey D. Ullman,Ravi Sethi, "Compilers – Principles, Techniques and Tools", Addison Wesley.

Reference Books:

1. D.M.Dhamdhere, "System software and Operating Systems", Second Revised Edition, Tata McGraw Hill, 1995.
2. D.M.Dhamdhere, "Introduction to System Software", Tata McGraw Hill, 1992.
3. Alfred V. Aho, Jeffrey D. Ullman, "Compiler Construction".

Course contents and Lecture schedule

S .No	Topics	No.of periods
1.	**Introduction and Machine architecture**	
1.1	Assemblers, loaders, linkers, macros, compilers, operating system, formal system	3
1.2	Simplified instructional computers – architecture, programming	6
2.	**Assemblers**	
2.1	Basic functions, algorithm , data structures	3

2.2	Machine dependent features	3
2.3	Machine independent features	4
2.4	Single pass and Multi pass assemblers	3
2.5	Implementation Examples	1
3.	**Loaders and Linkers**	
3.1	Basic functions	1
3.2	Machine dependent features	4
3.3	Machine independent features	2
3.4	Dynamic Linking, linkage editor and linking loader	3
3.5	Implementation Examples	1
4.	**Macro processors**	
4.1	Basic functions	2
4.2	Machine independent features	4
4.3	Recursive macro processors	2
4.4	General purpose macro processors	1
4.5	Macro processors integrated with language processors	1
4.6	Implementation Examples	1
5.	**Compilers**	
5.1	Analysis of source program (formal grammer)	2
5.2	Phases of compilers	2
5.3	Components of compilers	1
	Total	50

Course Designers

1. **B. Subbulakshmi bscse@tce.edu**
2. **B. Thanga Parvathi btpcse@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 34	3	-	-	3

C34 Operating Systems: Configuration and Use 3:0

Preamble: In the BE Computer Science and BE Information Technology program, the Course Operating Systems: Configuration and Use is offered as a common subject. One of the main assumptions made is that the course on "configuring and Using Operating Systems" would be studied from the viewpoint of the user-stakeholder and from a practical perspective of using the operating system.

In a course like this it is necessary to decide the "vehicle" OS through which the course will be conducted. In view of the growing base and popularity of the Open Source Linux, it is proposed to use it heavily. Windows XP Professional could be a viable choice, but the course could become dated when the Windows 7 OS is released shortly.

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, softwares and equipment to analyze problems.

k. Graduate who can participate and succeed in competitive examinations.

Competencies

1. Explain why the leading open source operating system Linux is powerful and popular in the community of users
2. Make an informed choice about the particular implementation, version, etc. for installation and use in a home or business setting
3. Install the Linux OS and make an initial configuration usable
4. Navigate GNOME and KDE to make the initial settings of the OS
5. Perform the various tasks of user management such as creating shell accounts, user accounts, restricted user accounts, software accounts, group user accounts, etc.
6. Create Linux file systems
7. Perform the various tasks of file system management such as configuring permissions, umask, groups, backing and restoring file systems, and navigating the file systems
8. Perform the tasks of package management dealing with compressed files
9. Perform GUI management, configure the GUI components, X server, etc.
10. Perform the various tasks of network management such as setting up IP addresses, configuring a domain, determining & applying subnets, configuring routing & masquerading, and setting up daemons

11. Configure and use Domain Name Services
12. Configure and make available the Internet mail services
13. Configure the Apache Web server for use
14. Setting up other services such as FTP, etc.
15. Use text processing tools such as vi and emacs
16. Write simple shell scripts to perform typical administrative tasks and simple applications

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	20
2	Understand	30	40	30
3	Apply	10	10	10
4	Analyze	20	10	10
5	Evaluate	10	10	20
6	Create	10	10	10

Course Level Learning Objectives:

Remember

1. List the various Linux Distributions
2. Which is the first Linux Distribution?
3. List the Linux Distributions Built on RedHat Linux Kernel
4. List the Linux Distributions Built on Debian Linux Kernal
5. What are the configuration tools used in RedHat Linux and SuSe Linux ?
6. Which is the most popular Linux distribution in the world?
7. List the Linux partitions for Linux Installation
8. Define a swap partition
9. What is meant by a Shell?
10. List the alternate web servers available for Linux
11. List Xwindow system commands and Configuration Files
12. List the Database Management Systems for Linux
13. Write the methods of managing user environments
14. List the file system Hierarchy Standards
15. What are the journaling file system types?
16. What are the entries in fstab files?
17. List the GNOME office applications
18. What is SQL Anywhere?

Understand

1. How to install Linux in a computer where Windows is already installed?
2. How to use the free space on the Hard Disk for Linux Installation?
3. Describe the fdisk utility ?
4. Explain the reasons for choosing the MBR/Boot sector/floppy disk for LILO
5. How is Authentication ensured when starting Linux?
6. How to run jobs in the background?
7. How chkconfig works?
8. How to login using anonymous FTP login?
9. Explain the available file system types
10. Explain the /proc File System
11. Explain about the Emacs and Vi Editors
12. Explain the GUI User Management tools?
13. How to Run MS Office on Linux?

Apply

1. Configure the Graphical system using XFree86 graphical software
2. Reconfigure the graphical system using command line
3. Configure the Apache web server
4. Configure the client browsers
5. Perform the postfix configuration
6. Perform the sendmail configuration
7. Set up and manage the MySQL database servers
8. Configure the C Shell and TCSH shell
9. Configure the Xorg Server
10. Upload an Image using F-Spot

Analyze

1. Differentiate the IDE and SCSI Interface partitions
2. Write the Disk Druid utility in Red Hat Linux
3. Analyze the cause, if Linux installation program does not start
4. Analyze the details of setting up the Linux Graphical Interface
5. Analyze the tasks performed on the system using syslog files
6. Analyze the changes made when Creating or deleting the new /etc/fstab entries
7. Analyze the mounting and unmounting of various devices using mount/umount commands
8. Analyze the system run levels in inittab
9. Analyze the configuration of Various Shells
10. Analyze the Special Features of Bash Shell

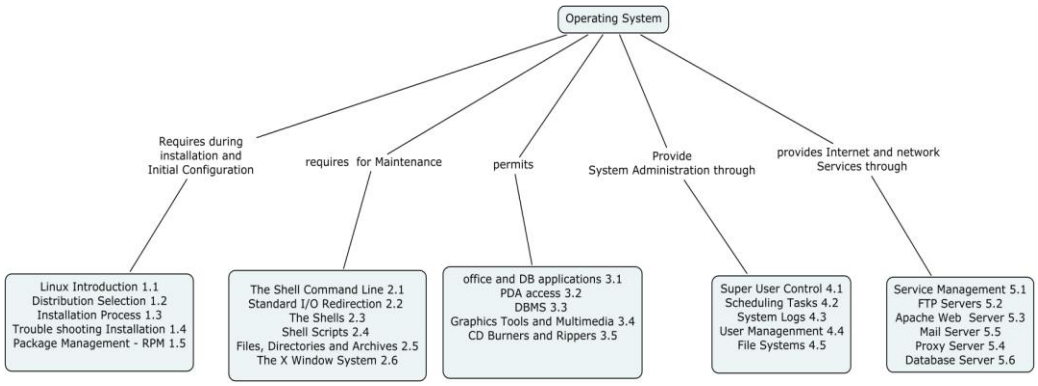
Evaluate

1. Compare the Drive letter assignments in Windows and Subdirectory mount points in Linux
2. Obtain the list of available system directories
3. Compare the Linux Desktop Editors
4. Compare the Graphics Tools for Linux
5. Compare the Multimedia and Sound Applications
6. Compare Video and DVD Projects and Applications
7. Compare the C shell and Z shell
8. Compare the Various Display Managers
9. Evaluate the Local Proxy Server Configuration
10. Compare the different Database Management Systems available on Linux

Create

1. Give the example mount points for a multiple partition installation
2. Create the boot disk to use when installing Linux
3. Create the default grub entry to Linux Operating system by editing the grub.conf file
4. Create file systems using various utilities
5. Create a new scheduled task using cron utilities
6. list the steps to install and uninstall RPM packages
7. Install software applications from compressed archive
8. Perform power and arithmetic operations using functions in openoffice.calc
9. Generate a chart in openoffice.writer using data in a openoffice.calc created by any equation.
10. Create a presentation about the college using Openoffice.impress
11. Create a New Image using GIMP

Concept Map:



Syllabus

Linux Installation: Introduction to Linux , Linux distributions , The Installation Process , Troubleshooting Installation ,Software package Management with RPM

Linux Shell, Files and Xwindow: The Shell command line , Standard I/O redirection , The various shells, Shell scripts , Linux Files Directories and Archives , The X window system **Linux Software Management**: Office and Database Applications , PDA access , DBMS , Graphics tools and Multimedia , CD Burners and rippers **System Administration**: Super user control , Scheduling tasks , System logs , User Management, File systems, **Internet and Network Services** Managing services , FTP Servers , Apache Web server, Proxy server , Mail Server , Database servers.

Textbooks:

1. Richard Petersen, The Complete Reference Linux Sixth Edition, Tata McGraw-Hill, 2008
2. Nicholas Wells, Guide to Linux Installation and Administration, Vikas Publishing House, 2000

Reference Book :

1. Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson, Lar Kaufman "Running Linux", Fourth Edition, December 2002

Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Linux Installation**	
1.1	Introduction to Linux	1
1.2	Linux distributions	1
1.3	The Installation Process	2
1.4	Troubleshooting Installation	1
1.5	Software package Management with RPM	1
2	**Linux Shell, Files and Xwindow**	
2.1	The Shell command line	1
2.2	Standard I/O redirection	1
2.3	The Shells	2
2.4	Shell scripts	2
2.5	Linux Files Directories and Archives	1
2.6	The X window system	1
3	**Linux Software Management**	
3.1	Office and Database Applications	2
3.2	PDA access	1
3.3	DBMS	2

3.4	Graphics tools and Multimedia	1
3.5	CD Burners and rippers	2
4	**System Administration**	
4.1	Super user control	2
4.2	Scheduling tasks	1
4.3	System logs	2
4.4	User management	1
4.5	File systems	2
5	**Internet and Network Services**	
5.1	Managing services	2
5.2	FTP Servers	1
5.3	Apache Web server	1
5.4	Proxy server	2
5.5	Mail server	2
5.6	Database servers	2
	Total No of Hours	40

Course Designers:

1. K. Sundarakantham kskcse@tce.edu
2. T. Subbulakshmi subbulakshmi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 35	3	-	-	3

C35 Design and Analysis of Algorithms 3:0

Prerequisite: Data Structures, Counting and Probability.

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) Analyze the asymptotic time complexity of algorithms

iii) Construct algorithms for solving engineering problems using various algorithm design paradigms.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

k. Graduates can participate and succeed in competitive examinations.

Competencies

1. Analyze the correctness of algorithms using induction and loop invariants.
2. Analyze the worst-case running time of algorithms using asymptotics and compare the asymptotic behaviors of different functions.
3. Analyze algorithms using amortized analysis (if needed) and Describe different methods employed for amortized analysis like the potential method and the accounting method. Recall the analysis of simple algorithms that requires this technique.
4. Describe the divide and conquer technique and infer when an algorithm design situation requires the use of this technique; Recall standard algorithms which use this technique; Construct new algorithms using this technique; Derive recurrences to characterize the running time of these algorithms and subsequently solve them.
5. Describe the dynamic programming and infer when an algorithm design situation requires the use of this technique; Recall standard algorithms which use this technique; Construct new algorithms using this technique and analyze their time and space complexities.
6. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recall algorithms that employ this paradigm. Construct greedy algorithms, and analyze them.
7. Recall the various sorting algorithms and their analysis. Construct algorithms that use sorting as a sub-procedure.

8. Demonstrate the understanding of algorithms which employs data structures and analyze how their performance is affected based on the choice of data structures. Construct new data structures by augmenting existing ones and leverage these in synthesizing new algorithms.
9. Describe major graph based algorithms and recall their analysis. Formulate engineering problems as graph problems (if appropriate) and construct graph based algorithms to solve these problems and subsequently analyze them.
10. Understand complexity classes P, NP, NP-Complete, NP-Hard and recall the representative problems under these complexity classes. Demonstrate an understanding of the notion of polynomial reductions and apply it to standard problems.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	20	10	10	10
2	Understand	30	20	20	20
3	Apply	40	40	30	40
4	Analyze	10	10	20	20
5	Evaluate	0	20	20	10
6	Create	0	0	0	0

Course Level Learning Objectives

Remember

1. Define the Big-oh notation (O) in asymptotics.
2. State the key properties which a problem should have so that it can be solved by a greedy algorithm.
3. State the use of a RAM model in analyzing algorithms.
4. Define the notion of a cut in a weighted undirected graph G(V,E)
5. State why the functions n and $n^{(1 + \sin n)}$ cannot be asymptotically compared.

Understand

1. Use Mathematical induction to show that if n is an exact power of 2, then the solution to the recurrence,

$$T(n) = \begin{cases} 2 & \text{if } n = 2, \\ 2T(n/2) + n & \text{if } n = 2^k, \text{for } k > 1 \end{cases}$$

is $T(n) = n \lg n$.

2. Illustrate Kruskal's algorithm for Minimum Spanning Tree (MST) construction with a Pseudo-code.

3. 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)).
4. 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"
5. Illustrate an algorithm which uses a dynamic programming technique, to solve the all-pair-shortest-paths problem, with the help of a pseudo-code snippet.
6. Illustrate the steps involved in finding the shortest path from a source vertex to all the other vertices in a Directed Acyclic Graph (DAG) with the help of an example.

Apply

1. Although merge sort runs in Θ(n lg n) worst-case time and insertion sort runs in Θ(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 Θ(nk) worst-case time.
   b. Show that the sub-lists can be merged in Θ(n lg (n/k) worst-case time.
2. Defend through a precise graph theoretic argument the following statement:
   "Let G(V, E) be a connected, undirected graph with a real-valued weight function defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let (S, V – S) be any cut of G that respects A, and let (u,v) be a light edge crossing (S, V – S). Then, edge (u, v) is safe for A"
3. Determine the optimal parenthesization of a matrix-chain produce whose sequence of dimensions is { 5, 10, 3, 12, 5, 50, 6}
4. Show how quick sort can be made to run in O(n lg n) time in the worst case
5. 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).

Analyze

1. Analyze the problem of selecting the i^{th} smallest element in a sequence of numbers using a greedy algorithm and infer the reason for selecting the size of the sub-problem/sub-sequence as greater than 4.
2. 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.
3. Compare and contrast Divide-and-Conquer paradigm with the Dynamic Programming paradigm and identify some representative problems to which each of these paradigms can be applied successfully.
4. Analyze whether the Travelling Sales Person problem (TSP) is decision reducible.

(i.e.) TSP $\equiv_T^P$ TSPD

5. Recall the following facts, proved 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.

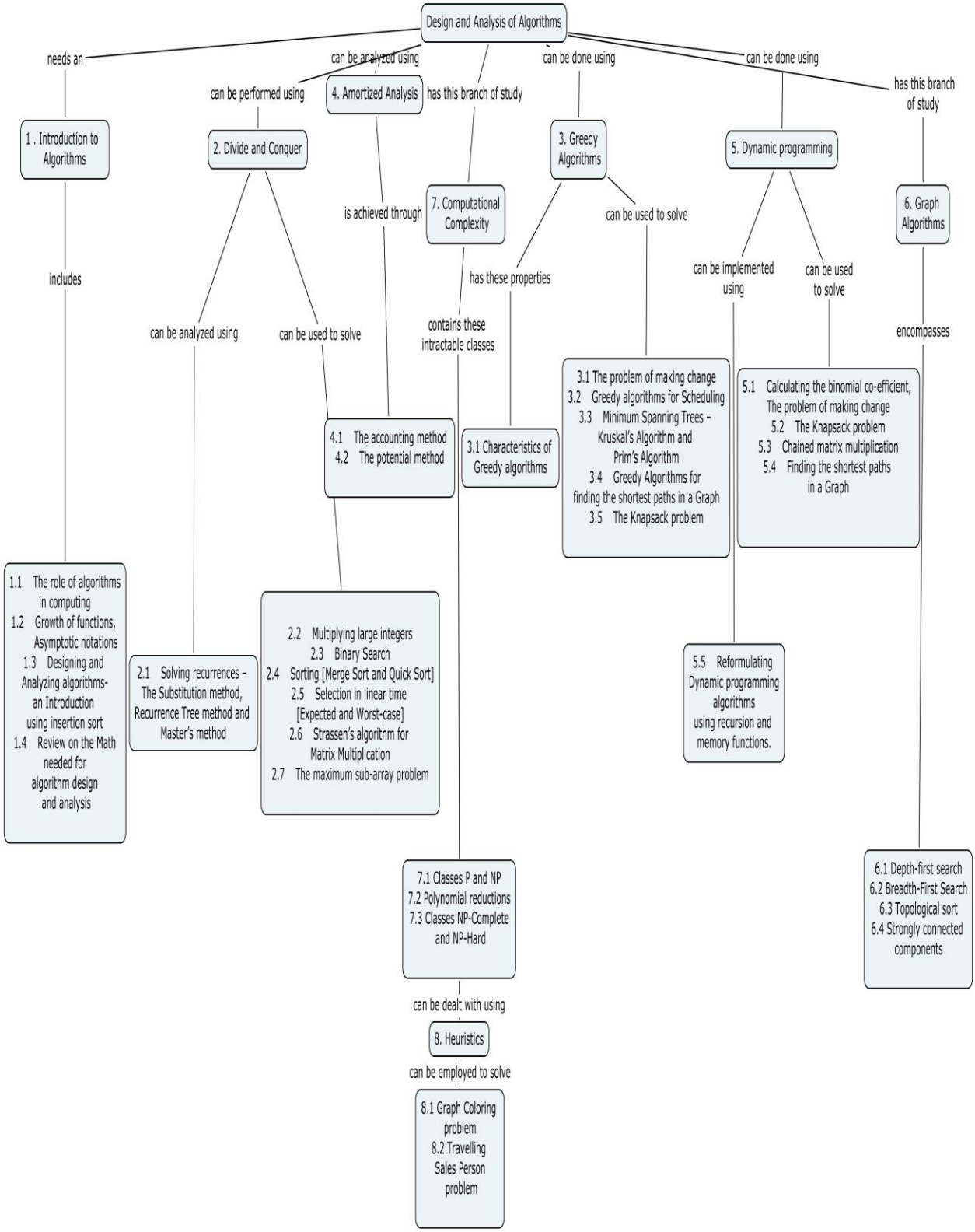
Evaluate

1. Evaluate the validity of the three cases of Master's theorem with an intuitive proof using recurrence trees.
2. Evaluate an asymptotically tight solution to the following recurrence using a recursion tree.

   $$T(n) = T(\alpha n) + T((1 - \alpha)n) + cn,$$

   where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.
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. Assess, which is a more efficient way to determine the optimal number of multiplications in a matrix-chain multiplication problem: enumerating all the ways of parenthesizing the product and computing the number of multiplications for each, or running the recursive matrix chain multiplication algorithm that was discussed in the class. Justify your answer.
5. Let X be an NP-Complete problem. Consider a decision problem Z ε NP such that X $\leq_T^P$ Z. Then defend that Z is also NP-Complete.

Concept Map


Design and Analysis of Algorithms
needs an
1 . Introduction to Algorithms
includes
1.1 The role of algorithms in computing
1.2 Growth of functions, Asymptotic notations
1.3 Designing and Analyzing algorithms- an Introduction using insertion sort
1.4 Review on the Math needed for algorithm design and analysis
can be performed using
2. Divide and Conquer
can be analyzed using
2.1 Solving recurrences - The Substitution method, Recurrence Tree method and Master's method
can be used to solve
2.2 Multiplying large integers
2.3 Binary Search
2.4 Sorting [Merge Sort and Quick Sort]
2.5 Selection in linear time [Expected and Worst-case]
2.6 Strassen's algorithm for Matrix Multiplication
2.7 The maximum sub-array problem
can be analyzed using
4. Amortized Analysis
is achieved through
4.1 The accounting method
4.2 The potential method
has this branch of study
7. Computational Complexity
contains these intractable classes
7.1 Classes P and NP
7.2 Polynomial reductions
7.3 Classes NP-Complete and NP-Hard
can be dealt with using
8. Heuristics
can be employed to solve
8.1 Graph Coloring problem
8.2 Travelling Sales Person problem
can be done using
3. Greedy Algorithms
has these properties
3.1 Characteristics of Greedy algorithms
can be used to solve
3.1 The problem of making change
3.2 Greedy algorithms for Scheduling
3.3 Minimum Spanning Trees - Kruskal's Algorithm and Prim's Algorithm
3.4 Greedy Algorithms for finding the shortest paths in a Graph
3.5 The Knapsack problem
can be done using
5. Dynamic programming
can be implemented using
5.5 Reformulating Dynamic programming algorithms using recursion and memory functions.
can be used to solve
5.1 Calculating the binomial co-efficient, The problem of making change
5.2 The Knapsack problem
5.3 Chained matrix multiplication
5.4 Finding the shortest paths in a Graph
has this branch of study
6. Graph Algorithms
encompasses
6.1 Depth-first search
6.2 Breadth-First Search
6.3 Topological sort
6.4 Strongly connected components

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, 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

Sl.no	Topic	Hours
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	4
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	2
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	2
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	1
6.2	Breadth-First Search	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	**40**

Course Designer:

1. Mr. Karthick Seshadri skcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 36	3	1	-	4

C36 Object Oriented Programming 3:1

Preamble: This course is offered in the third semester for the students of Computer Science and Engineering. As prerequisites, the courses such as C24: Computers and Programming, C28: Computer Programming lab has been offered in the previous semesters. .

Program Outcomes addressed

a. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, and mathematics and science problems.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

Competencies

At the end of the course, the student will be able to

1. Explain the basics of object oriented programming.
2. Explain the need for modular programs and describe the ways of achieving it.
3. Describe the concept of data encapsulation and appreciate the value of object orientation.
4. Explain software reusability through inheritance and to understand the notion of polymorphism.
5. Discuss the ways to watch for, indicate and handle exceptions.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3/End-semester examination**
1	Remember	30	20	20
2	Understand	30	30	20
3	Apply	40	50	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. What is encapsulation?
2. What is object oriented programming?
3. What is the difference between a static local variable and a global variable?
4. Indicate which of the following is the function prototype, the function header and the function call:
5. void showNum(float num)
6. void showNum(float);

Understand

1. What are the benefits of object oriented programming?
2. What is the purpose of friend functions?
3. What kinds of values may be specified as a default arguments?
4. Describe the purpose of a constructor..
5. List the advantage and disadvantage of using a private member function.
6. Discuss why converting a base class pointer to a derived class pointer is considered dangerous by the compiler?

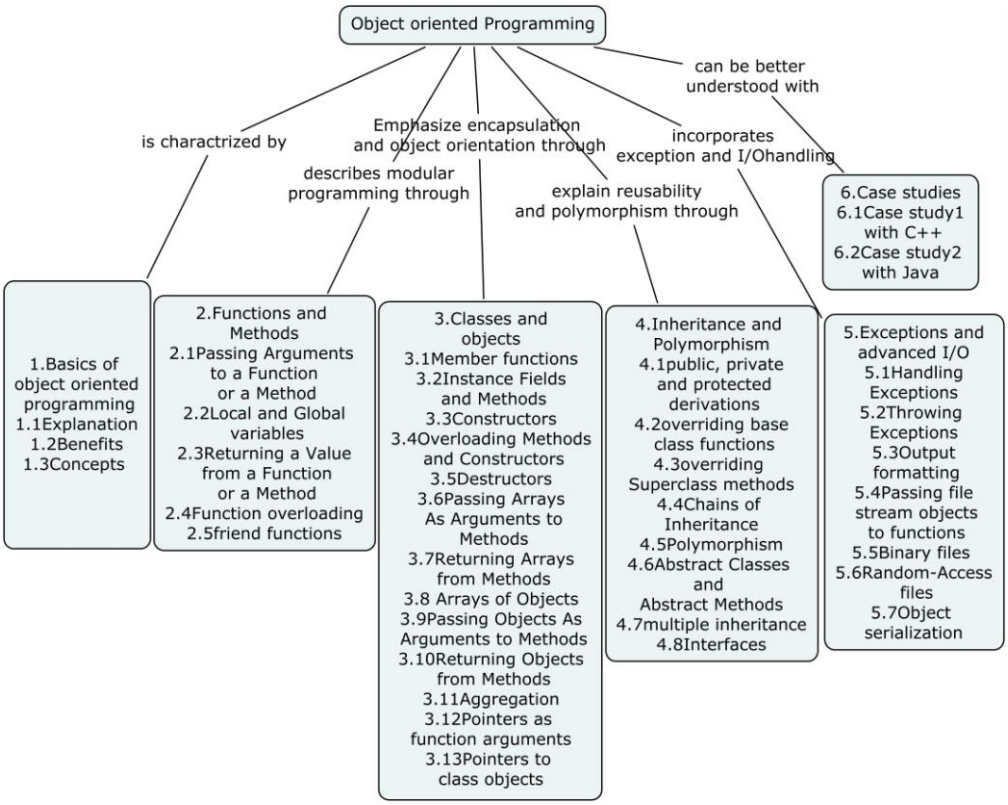
Apply

1. Write a complete C++ program with the two alternate functions specified below, of which each simply triples the variable count defined in main. Then compare and contrast the two approaches. These two functions are
   - Function tripleCallByValue that passes a copy of count call-by-value, triples the copy and returns the new value.
   - Function tripleByReference that passes count with true call-by-reference via a reference parameter and triples the original copy of count through its alias(i.e.,the reference parameter)
2. Write a program to calculate the area of a square and a rectangle using function overloading.
3. Write a Java class called Contains. This class defines a method called contains(), which takes as parameters an array of int called **a** as well as an int called **x,** and returns a Boolean which is true if and only if x occurs in a , false otherwise.
4. Given the following class, called NumberHolder, write some code that creates an instance of the class, initializes its two member variables, and then displays the value of each member variable.

```
public class NumberHolder {
  public int anInt;
  public float aFloat;        }
```

5. Write class Animal with the following properties:
   Private data members: familyType (Cat: Tiger, Lion…, Dog: Wolf, jackal…), weight.
   Two constructors, one is the default.
   get and set methods for each data member.
   toString method to print the data.
6. Write class Pet that derives from class Animal with additional properties:
   Data members: petName, petColor
   Two constructors, one is the default.
   get and set methods for new data members.
   toString method to print data of the pet.

Concept Map:



Syllabus:

Basics of Object oriented Programming- What is Object oriented programming? - Benefits of Object oriented programming - Object oriented programming principles – **Functions and Methods** - Passing Arguments to a Function or a Method – Local and Global variables - Returning a Value from a Function or a Method – Function overloading – friend functions - **classes and objects** – Member functions - Instance Fields and Methods - Constructors - Overloading Methods and Constructors – Destructors - Passing Arrays As Arguments to Methods - Returning Arrays from Methods - Arrays of Objects - Passing Objects As Arguments to Methods - Returning Objects from Methods – Aggregation – Pointers as function arguments – Pointers to class objects – **Inheritance**

and Polymorphism - public, private, and protected derivations – overriding base class functions – overriding Superclass methods - Chains of Inheritance – Polymorphism - Abstract Classes and Abstract Methods - multiple inheritance – Interfaces – **Exceptions and advanced I/O** - Handling Exceptions - Throwing Exceptions – Output formatting - Passing file stream objects to functions – Binary files – Random-Access files – Object serialization – **Case studies** - Case study1 with C++ - Case study2 with Java.

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:

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:

S.No.	Topic	No. of Lectures
1	**Basics of Object oriented Programming**	
1.1	Object oriented programming-explanation	1
1.2	Benefits of Object oriented programming	1
1.3	Object oriented programming concepts	1
2	**Functions and Methods**	
2.1	Passing Arguments to a Function or a Method	1
2.2	Local and Global variables	1
2.3	Returning a Value from a Function or a Method	1
2.4	Function overloading	1
2.5	friend functions	1
3	**Classes and Objects**	
3.1	Member functions	1
3.2	Instance Fields and Methods	1
3.3	Constructors	1
3.4	Overloading Methods and Constructors	1

S.No.	Topic	No. of Lectures
3.5	Destructors	1
3.6	Passing Arrays As Arguments to Methods	1
3.7	Returning Arrays from Methods	1
3.8	Arrays of Objects	1
3.9	Passing Objects As Arguments to Methods	1
3.10	Returning Objects from Methods	1
3.11	Aggregation	1
3.12	Pointers as function arguments	1
3.13	Pointers to class objects	1
4	**Inheritance and Polymorphism**	
4.1	public, private, and protected derivations	1
4.2	overriding base class functions	1
4.3	overriding Superclass methods	1
4.4	Chains of Inheritance	1
4.5	Polymorphism	1
4.6	Abstract Classes and Abstract Methods	1
4.7	multiple inheritance	1
4.8	Interfaces	1
5	**Exceptions and advanced I/O**	
5.1	Handling Exceptions	1
5.2	Throwing Exceptions	1
5.3	Output formatting	1
5.4	Passing file stream objects to functions	1
5.5	Binary files	1
5.6	Random-Access files	1
5.7	Object serialization	1
6	**Case Studies**	
6.1	Case study1 with C++	2
6.2	Case study2 with Java	2
	Total	**40**

Course Designer:

1. J. Jane Rubel Angelina janerubel@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 37	-	-	1	1

C37 Operating Systems and System Software Lab 0:1

List of Experiments

1. **Installation**
   Given a Hard disk, no CD-ROM, USB, Install an OS like Linux, Windows
2. **Assembling and Installing**
   Given RAM, Hard disk, Motherboard, VGA Card, processor, Bus, CPU assemble the system and install the first Operating System in the hard disk.
3. **LAN Setup – Windows**
   Setting up a LAN between two systems using cross cable in Windows (without using switches)
4. **FTP Configuration**
   Configure ftp, file transfer between the two systems
5. **LAN Setup – Heterogeneous OS**
   Heterogeneous OS in Setting up a LAN between two systems using cross cable in Windows using samba server (without using switches)
   a. Between Linux-> Windows
   b. Between Windows-> Linux
   c. Between Linux-> Linux (NIS & NPS)
6. **Apache Installation and Configuration**
   Install and Configure Apache Server and design and show a web page about the college
7. **squid Configuration**
   Install and Configure squid in a system.
8. **GIMP**
   Design a Poster using GIMP
9. **Configuration of Exim/sendmail mail client**
   Install and configure any one mail client
10. **Use of System calls and Symbol Table Creation**
    Use of any 10 system calls and create symbol table
11. **Assembler**
    Design a Single and Multi pass assemblers.
12. **Implementation of Loader**
    Implement the absolute and relocatable loader
13. **Implementation of linker**
    Implement the linker module

14. Implementation of lexical Analyzer

Implement the lexical analyzer to separate the tokens

Number of Batches: 10

Number of Students in Each Batch: 4

Software requirement Windows and Linux (debian, redhat, ubuntu)

Hardware requirement for each experiment: 1 Client Machine

For Experiment 5 & 7 server with (Linux OS)

Course Designers

1. Ms.K.Sundarakantham kskcse@tce.edu
2. Ms.T.Subbulakshmi subbulakshmi@tec.edu
3. Ms.B.Subbulakshmi bscse@tce.edu
4. Ms.B.Thangaparvathy btpcse@tce.edu

Sub. Code	Lectures	Tutorial	Practical	Credit
C38	-	-	3	1

C38 Object Oriented Programming Laboratory 0:1

Preamble
A dynamic learn-by-doing focus that encourages students to develop an application using object oriented approach.

Prerequisite: C36

List of Exercises:

1. C++ Program for passing arguments to a function(by value and by reference) and returning value from a function
2. Java Program for passing arguments to a method and returning value from a method
3. C++ Program for function overloading, constructor overloading, default arguments and use of non-inline functions
4. Java Program for method overloading and constructor overloading
5. C++ Program for passing arrays and objects as arguments to function and returning arrays and objects from functions
6. Java Program for passing arrays and objects as arguments to methods and returning arrays and objects from methods
7. C++ program for inheritance and overriding base class functions.
8. Java program for inheritance and overriding super class methods.
9. C++ program for abstract base classes and pure virtual functions.
10. Java program for abstract classes and abstract methods.
11. C++ program for Exception handling and File handling.
12. Java program for Exception handling and File handling.

Sample Applications:

1. Health care automation.
2. Employee expense management system.
3. Retail billing system.
4. Stock market trading System.
5. Code version repository.
6. Inventory management system.
7. Traffic management system.
8. Portfolio management system.
9. Tender management system.
10. Law and order automation.

Note: Students have to pick an application from the list of applications provided by the instructor and perform experiments 1 to 12 in accordance with the application picked.

Course Designer:

1. Mrs. J. Jane Rubel Angelina janerubel@tce.edu

39th meeting of the Academic Council Resolution No.39:10

Annexure-H

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

FOURTH SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2000 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

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

Board of studies Meeting on 31.10.2009 Approved in 39th Academic Council Meeting on 25.11.2009

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the year 2013-2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

Board of studies Meeting on 31.10.2009 Approved in 39th Academic Council Meeting on 25.11.2009

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014 onwards)

FOURTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
C 41	Discrete Mathematics	BS	4	-	-	4
C 42	Computer Networks: Use & Configuration	DC	3	-	-	3
C 43	Principles of Compiler Design	DC	3	-	-	3
C 44	Operating Systems: Principles and Design	DC	3	1	-	4
C 45	Software Engineering	DC	3	1	-	4
C 46	Data Bases: Practice	DC	3	1	-	4
PRACTICAL						
C 47	Network Programming Lab	DC	-	-	3	1
C 48	Data Bases Lab	DC	-	-	3	1
C49	Professional Communication	HSS	1	1	1	2
Total			**20**	**4**	**7**	**26**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
DC : Department Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS

(For the candidates admitted from 2013 - 2014 onwards)

FOURTH SEMESTER

S.No.	Sub. code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C 41	Discrete Mathematics	3	50	50	100	25	50
2	C 42	Computer Networks: Use & Configuration	3	50	50	100	25	50
3	C 43	Principles of Compiler Design	3	50	50	100	25	50
4	C 44	Operating Systems: Principles and Design	3	50	50	100	25	50
5	C 45	Software Engineering	3	50	50	100	25	50
6	C 46	Data Bases: Practice	3	50	50	100	25	50
PRACTICAL								
7	C 47	Network Programming Lab	3	50	50	100	25	50
8	C 48	Data Bases Lab	3	50	50	100	25	50
9.	C49	Professional Communication	3	50	50	100	25	50

* Continuous Assessment evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub code	Lectures	Tutorial	Practical	Credit
C41	4	-	-	4

C41 Discrete Mathematics **4:0**

Preamble:

An engineering student needs to have some basic mathematical tools and techniques. 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 the theory and applications of Set theory, logic and Automata theory.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of Mathematics, Science and Engineering.

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

c. Graduates will develop confidence for self education and ability for life-long learning.

Competencies

At the end of the course the student should be able to

1. Check the validity of the arguments.
2. Understand how to construct correct mathematical arguments.
3. Design of computers and electrical circuit.
4. Check whether a particular combination of words is a valid sentence or not
5. Specify a well defined set of rule of syntax by which certain formal Languages such as programming languages can be constructed.

Assessment Pattern

	Bloom's category	Test 1	Test 2	Test 3 / End Semester Examinations
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70
4	Analyze	0	0	0
5	Evaluate	0	0	0

6	Create	0	0	0

Course level learning objectives
Remember

1. Define a biconditional statement and draw its truth table
2. Show that $(P \wedge Q) \rightarrow (P \vee Q)$ is a tautology without constructing truth table.
3. Define term NAND
4. Show that following implication $[P \rightarrow (Q \rightarrow R)] \Rightarrow [(P \rightarrow Q) \rightarrow (P \rightarrow R)]$
5. Define minterm and maxterm.
6. Show that $A - B = A \cap B'$.
7. Show that by using mathematical induction $2+2^2+2^3+--------+2^n=2^{n+1}-2$
8. Define NFA with ε-moves.
9. Define a Moore machine.
10. List the steps in conversion a CFG to Chomsky normal form.

Understand

1. Estimate the PCNF and PDNF of the formula given by $(7P \rightarrow R) \wedge (Q \rightarrow P)$
2. Discuss this $(7Q), P \rightarrow Q, P \vee R \Rightarrow R$ by indirect method
3. Show that $S \vee R$ is tautologically implied by $(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow S)$
   using automatic theorem proving
4. 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
5. Let A,B,C are any three sets Prove that $(A \cup B)XC = (AXC) \cup (BXC)$.
6. 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.
7. Discuss NFA for regular expression $(01+10)^+$
8. Discuss the language generated by $S \rightarrow aB, B \rightarrow AB, aA \rightarrow b, A \rightarrow b, B \rightarrow Aa.$
9. Construct Pushdown automata which accept the language with equal number of a's and b's with empty store.

Apply
1.Show that $7(p \wedge (7q \wedge r) \vee (q \wedge r) \vee (p \wedge r) \Leftrightarrow r$
2. Define NAND and NOR. Prove that the connectives NAND and NOR are commutative.

3. Show that $S \vee R$ is tautologically implied by $(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow S)$

4. Show that $(x)(P(x) \vee Q(x)) \Rightarrow (x)P(x) \vee (\exists(x))Q(x)$,Using indirect method.

5**.** Prov**e** that L= $\{0^{i^2}$ /i is an integer,i>1$\}$ is not regular

6.Discuss the Pushdown automata .

Syllabus

Logic

Introduction – Statements and Notations , Negation – Conjuction – 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 calculas

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 Formulas – 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.

Automata Theory

Finite State machines and Basic Definitions, Non-Deterministic Finite Automata, Finite Automata with ε - moves, Regular Expressions, Two way Finite Automata, Finite Automata with output

Grammar

The Pumping Lemma for Regular Sets, Closure Properties of Regular Sets, Context – Free Grammar and its Simplification, Derivation Trees, Chomsky-Normal Form and Greibach Normal Form

Text Books

1. T.P.Tremblay and R.Manohar ,"Discrete Mathematical Structures with application to Computer Science",Tata McGraw Hill,1997

2. John.E.Hopcorff and Jeffery D. Ullman ,"Introduction to Automata Theory, Languages and Computation"Narosa Publishing House,1997.

References

1. Dr. M.K.Venkataraman.,Dr.N.Sridharan and N.Chandrasekaran, Discrete Mathematics,National Publishing Company,Chennai.of India(2004)

Course content and lecture schedule

Sl.No.	Topics	No.of Lectures
	Unit I	
1	Introduction – Statements and Notations	1
2	Negation – Conjuction – Disjunction – Truth table	1
3	Conditional – Biconditional – Tautological Statements – Equivalence of Formulas	2
4	Tutorial – I	1
5	Duality Law – Tautological Implications-Functionally Complete set of Connectives –Other Connectives	1
6	Two State Devices & Statement Logic	1
7	Disjunctive – Conjunctive – Principle Disjunctive – Principle Conjunctive	2
8	Tutorial – II	1
	UNIT II	
11	Checking the validity using the truth table, Rules of Inference –Consistency of premises and Indirect Method	2
12	Automatic Theorem proving	1
13	Predicates-Function, Variables and Quantifiers	1
14	Tutorial – I	1
15	Predicate formulas – Free and Bound Variables	1
16	Valid Formulas – Equivalences	1
17	Theory of Inferences for the predicate Calculus	2
18	Tutorial – II	1
	UNIT III	
19	Basic Definitions – Operations on Sets –Identities	1
20	Ordered pairs & n-tuples –Cartesian Products	1
21	Definition of Relation –Binary Relation – Properties-Matrix - Graph	2

22	Tutorial - I	1
23	Equivalence relation – Compatibility Relation- Composition of relation- Poset	2
24	Functions –Composition-Inverse	2
25	Tutorial - II	1
	UNIT IV	
26	Finite State machines and Basic Definitions	2
27	Non-Deterministic Finite Automata	1
28	Finite Automata with ε - moves	1
29	Tutorial – I	1
30	Regular Expressions	2
31	Two way Finite Automata	1
32	Finite Automata with output	1
33	Tutorial – II	1
	UNIT V	
34	The Pumping Lemma for Regular Sets	2
40	Closure Properties of Regular Sets	1
41	Context – Free Grammar and its Simplification	2
42	Tutorial – I	1
43	Derivation Trees	1
44	Chomsky-Normal Form and Greibach Normal Form	2
45	Tutorial – II	1

Course Designers

1. V.Mohan vmohan@tce.edu
2. G.Jothilakshmi gjlmat@tce.edu
3. A.P.Pushpalatha appmat@tce.edu
4. K.Angaleeswari kangaleeswari@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 42	3	-	--	3

C42 COMPUTER NETWORKS: USE AND CONFIGURATION 3:0

Preamble: Computer network use and configuration is the discipline which studies the theoretical, practical and managerial aspects of designing and managing computer networks. The course will enable the students to familiarize the various aspects of computer networks such as what they are, how they work, how to design, build and configure them.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.
b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.
c. Graduate will demonstrate skills to use modern engineering tools, softwares and equipment to analyze problems

Competencies

1. Explain the role of different Layers of OSI Network model
2. Determine the performance of a given Computer network
3. Describe the components of Computer networks
4. Synthesize Addressing mechanisms for Computer networks
5. Design a network for a given set of specifications
6. Design services based on Computer networks

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	10	10
2	Understand	30	20	20
3	Apply	30	40	30
4	Analyze	20	30	20
5	Evaluate	0	0	0
6	Create	0	0	20

Course Level Learning Objectives

Remember:

1. List out the disadvantages of optical fiber.
2. Describe the need and use of modems.
3. What are the advantages of FDDI over a basic Token Ring?

4. What is circuit switching and what are its chief characteristics?
5. Describe the encapsulation involved in the creation of an Ethernet frame.
6. Differentiate the basic operation of hubs over switches.
7. What is the difference between routable and non- routable protocols?
8. What is multicast routing?
9. What is a DNS resource record?
10. List the port numbers associated with FTP, SMTP, and HTTP.

Understand:

1. How are the guided media differing from unguided transmission media?
2. Identify the address class of 123.167.23.20 and 250.10.24.96.
3. In a mesh network, how many connections are required to connect 20 computers?
4. In a circuit-switched network, can multiple circuits share a single optical fiber?
5. Compare 10Base5, 10 Base2, 10BaseT, and 10Base F mediums used in IEEE 802.3.
6. Explain the data frame format of IEEE 802.5 Standard?
7. Write the subnet, broadcast address and valid host range for the following:

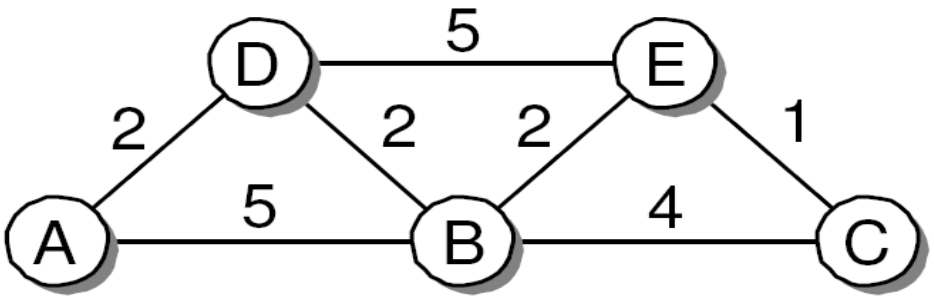
1. 172.16.10.5	255.255.255.128
2. 172.16.10.33	255.255.255.224
3. 172.16.10.65	255.255.255.192
4. 172.16.10.17	255.255.255.252

8. Identify a network as Class A, B, or C, given an IP address and a network mask.
9. The Network address is 192.168.10.0 and the Subnet mask is 255.255.255.252. Find out
   1. How many subnets?
   2. How many hosts?
10. List the 3 non-routable address spaces and explain what makes them non-routable
11. What is anonymous FTP and why would you use it?

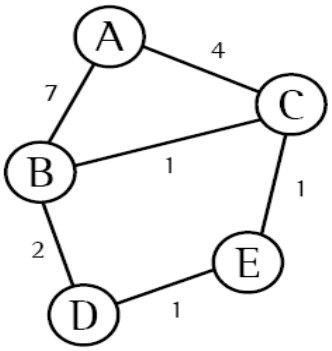
Apply

1. If a digital signaling system required to operate 9600bps and encodes 16 bit word, compute the minimum bandwidth required for this channel.
2. For the bit stream 100010100, draw the waveform for Manchester and Differential Manchester Coding and also discuss the advantages of the coding schemes
3. If a binary signal (two sampling levels) is sent over a 3-kHz channel whose signal-to-noise ratio is 20 dB, compute the maximum achievable data rate?
4. Calculate the latency of a packet transferred between two hosts A and B on a local network, where they are connected via a cable of length 6.21 m., the packet size is 2024 bytes, and the capacity of the cable is 56 bps. (Assume that queuing delays are not considered. Also recall that 1 byte = 8 bits, and the speed of light = 3.0x108 m/s.)

5. An end system sends 50 packets per sec using UDP protocol over a full duplex 100Mbps Ethernet LAN connection. Each packet consists 1600 bytes of Ethernet frames payload data. Compute the throughput at UDP layer?
6. Convert a classless (CIDR) network address (e.g. 192.168.0.0/24) to its IP address/mask equivalent (e.g. 192.168.0.0 255.255.255.0) and vice versa
7. A collection of five routers is to be connected in a point-to-point subnet. Between each pair of routers, the designers may put a high-speed line, a medium-speed line, or a low-speed line, or no line. If it takes 100 ms of computer time to generate and inspect each topology, compute the time required to inspect all of them.
8. The Internet is roughly doubling in size every 18 months. Approximate estimates put the number of hosts on it at 7 million in January 1996. Use this data, compute the expected number of Internet users in the year 2008.
9. Compute the shortest paths from router A to any of the other routers in the figure by means of Dijkstra's shortest-path algorithm.

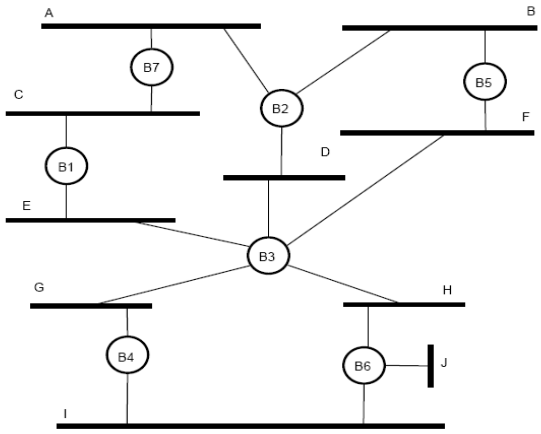


10. Draw a simple topology of routers with multiple connections. The connections should be assigned costs. What is the state of the routing table for each node before any distance vectors were exchanged? Now, exchange a few distance vectors between the routers and determine the routing tables which have changed.



Analyze:

1. Identify whether circuit switching system or a packet switching preferable for broadcasting a copy of a video presentation? Why?
2. Given the extended LAN shown in the figure below, indicate which ports are not selected by the spanning tree algorithm. The LAN's in the figure are labeled A-J and the bridges in the figure are labeled B1-B7. Hub Bi has an ID of i which is used as the tie breaker.



3. Switching can improve the efficiency of a network's link utilisation, but may also cause problems. In a packet-switched network, two particular problems are *increased latency* and *data loss*.

   (*i*) Taking the IP packet-switching, explain how latency and loss might occur.

   (*ii*) To what extent are the problems of latency and loss less significant in circuit-switched networks?

4. The switching process consists roughly of a *demultiplexing* stage, a *routing* stage and a *remultiplexing* stage. For each of the following examples of switching, explain what is being demultiplexed, what routing decisions are made, and how remultiplexing is performed:

   (*i*) packet switching in the postal network;

   (*ii*) packet switching in an Ethernet switch;

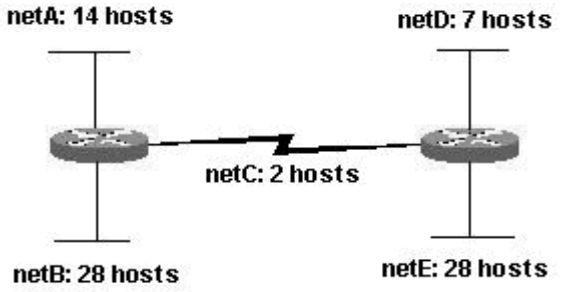
   (*iii*) packet switching in an IP router;

   (*iv*) circuit switching in the telephone network;

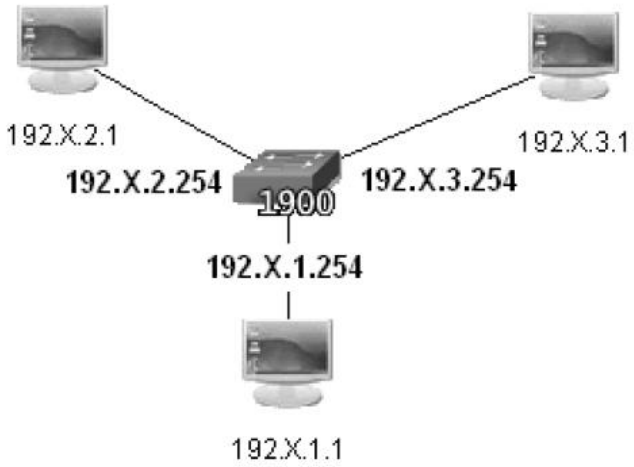
5. Create a VPN to connect to branch office of the department. What would be the preliminary requirement?

Create:

1. Given the Class C network of 204.15.5.0/24, subnet the network in order to create the network in Figure with the host requirements.

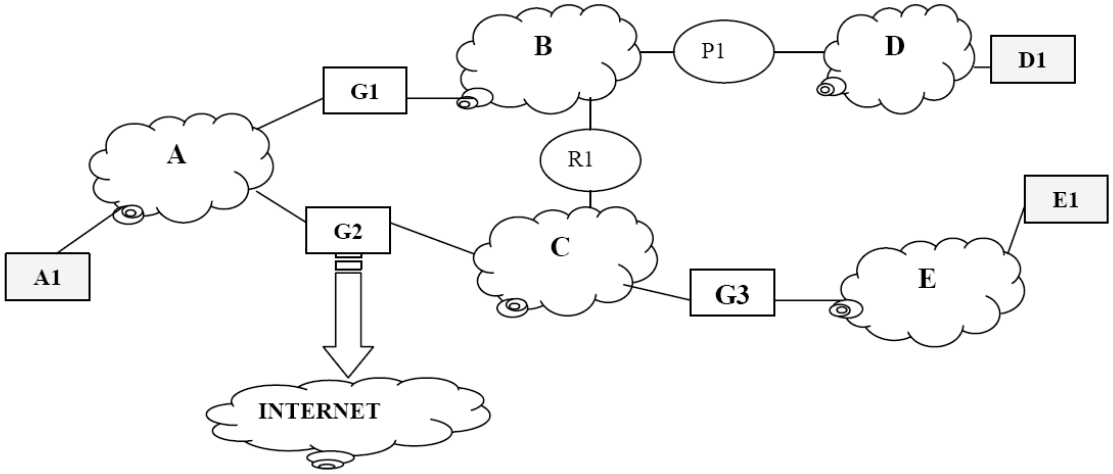


2. You have 3 computers belonging to different networks. Configure them to be able to access each other. Note: Use at least three routers, one for each internal network. Change the IP Address to match the diagram below. Take note that you will need a fourth network (a new network address!) for all the routers so that they can communicate with one another. You can also use switches.



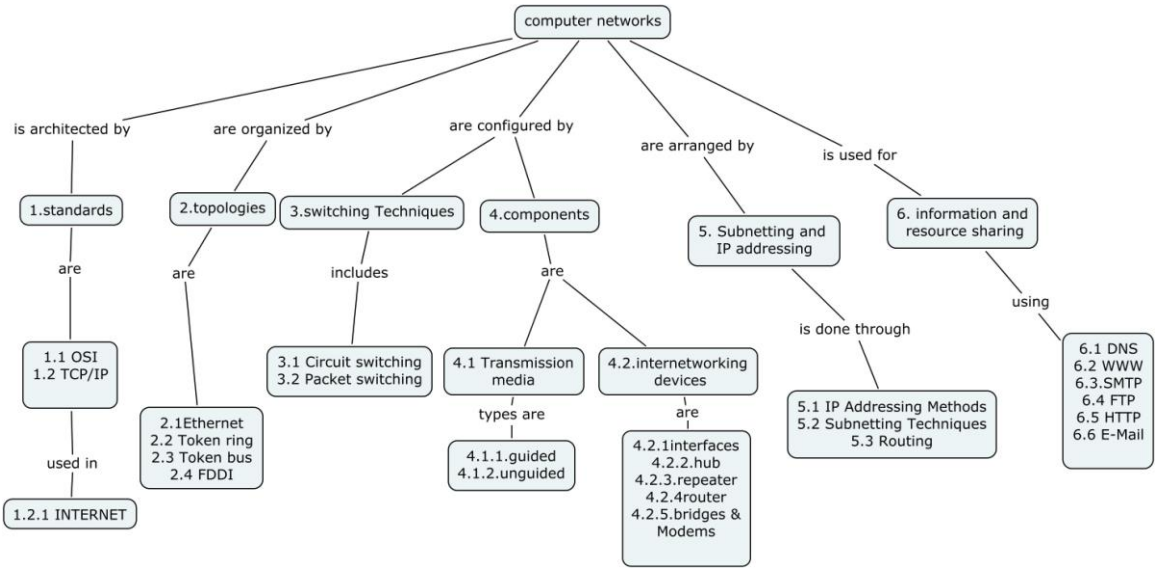
3. A routing protocol is a system used by routers to automatically maintain their forwarding tables. Outline a simple routing protocol which might be used to maintain the table under a *shortest path* routing policy. Mention any additional information that you must store in the router, and any problems you notice.
4. A company has two LANs, one in Chennai with 300 hosts and another one in Madurai with 150 hosts. Could it be possible to connect those networks to the Internet using only one Class C network addresses? Justify the answer. If the answer is positive, create a network layout, assign IP addresses to every router and to one host in the network, and specify the routing tables of all routers and the specified host.
5. In the figure below there is a set of Ethernet LANs (A,B,C,D and E) that conform to the Intranet policies of one company. The LANs are interconnected with three routers (G1, G2 y G3), a bridge (P1) and a hub (R1). The Internet connection is managed by router G2. The Company has only one class C network address for this Intranet. We

highlight only three hosts - A1, D1 and E1. Assume that there are more hosts in each LAN.



Assign IP addresses to every network element (network IDs, hosts, etc.). Specify the routing tables of all routers and the one corresponding to host D1.

Concept Map



Syllabus:

Introduction: Data Communication Concepts, Data Encoding **Network Standards**: OSI Architecture, TCP/IP, Internet Architecture **LAN Topologies**: Ethernet - Token Ring, Token Bus, FDDI **Switches:** Circuit Switching, Packet Switching **Components:** Transmission Media (Guided and unguided media) Internetworking Devices: Interfaces, Hub, Repeaters, Bridges, Routers and Modems **Subnetting and IP Addressing:** IP Addressing and Methods, Sub netting - Routing – Distance Vector Routing - Link State Routing, **Information and Resource sharing:** DNS, WWW, SMTP, FTP,HTTP, E-Mail

Textbook

1. Behrouz A.Forouzan, "Data Communication and Networking", Tata McGraw-Hill, 2004.

Reference Book

1. L.Peterson and Peter S.Davie, "Computer Networks", Harcourt Asia Pvt.Ltd., Second Edition.
2. Andrew S.Tanenbaum,"Computer Networks",PHI,Fourth Edition,2003.
3. Peterson and Davie, Computer Networks: A Systems Approach,3rd Edition, Morgan Kaufmann, 2003
4. Black U., "Computer Networks- Protocols, Standards and Interfaces", PHI.

Course Contents and Lectures schedule

No.	Topics	No of Lectures
1.	**Introduction**	1
1.1	**Standards:** OSI Architecture	2
1.2	TCP/IP	2
1.3	Internet Architecture	2
2	**Topologies**	
2.1	LAN-Ethernet	2
2.2	Token Ring	2
2.3	Token Bus	2
2.4	FDDI	2
3	**WAN: Switches**	
3.1	Circuit switching	2
3.2	Packet switching	2
4	**Transmission Media**	
4.1.1	Guided Media	2
4.1.2	Unguided Media	2
4.2	**Internetworking Devices**	
4.2.1	Interfaces	1
4.2.2	Hub	1

4.2.3	Repeaters	1
4.2.4	Bridges	1
4.2.5	Routers and Modems	2
5	**Subnetting and IP Addressing**	
5.1	IP Addressing and Methods	1
5.2	Sub netting	2
5.3	Routing: Distance Vector Routing - Link State Routing	2
6	**Information and Resource Sharing**	
6.1	DNS	1
6.2	WWW	1
6.3	SMTP	1
6.4	FTP	1
6.5	HTTP	1
6.6	Email	1
	Total	40

Course Designers:

1.S. Srinivasan **ssnit@tce.edu**

2.S. Muthuramalingam **smrit@tce.edu**

3.S. Gilbert Nancy **gilbertnancyl@tce.edu**

4.P.Subathra **pscse@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 43	3	-	-	3

C43 Principles of Compiler Design **3:0**

Preamble: Prerequisites are **C33** System Software**,** C23 – Computer Organization and Design. This course gives an overview of the various phases of compiler and explains how a high level program is converted to a machine language program.

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

e. Graduates will demonstrate an ability to visualize and work on laboratory and multidisciplinary task.

Competencies

1. Explain the role of each phases of a compiler with its construction tools.
2. Implement various parsers like top down, bottom up, operator precedence parsers.
3. Understand the importance of intermediate code generation and code optimization.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	10	10
2	Understand	50	20	20
3	Apply	30	70	70
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level Learning Objectives

Remember

1. List the criteria to the selection of optimizing transformations.
2. Define induction variable. Give examples.
3. Explain the format of activation record.
4. Define Compiler. What are the phases of the Compiler? Explain with a neat diagram.

5. What are Compiler Construction Tools? Explain its specifications in detail.
6. What are the error recovery actions in a lexical analyzer?
7. What are the algebraic properties of regular expressions?

Understand

1. Define the role of input buffer in lexical analysis.
2. Write the function of syntax directed translation.
3. Write three address code to the statement A=-B*(C+D).
4. What do you mean by code motion? Explain.
5. What is basic block? Explain the role of flow graph in basis blocks
6. How would you calculate the cost of an instruction?
7. How would you map names to values?
8. What does operator-precedence parser do to the input id + id * id.

Apply

1. Write regular expression to generate identifiers give examples.
2. Construct of a parse tree to the string-(id+id) using left most derivation.
3. Draw transition diagrams to floating point numbers.
4. Write procedure for the construction of NFA from a regular expression
5. Discuss neatly the language for specifying lexical analyzers.
6. Construct NFA, DFA for the expression aa* ¦ bb*
7. What is SLR parsing .construct SLR parsing table for the grammar.

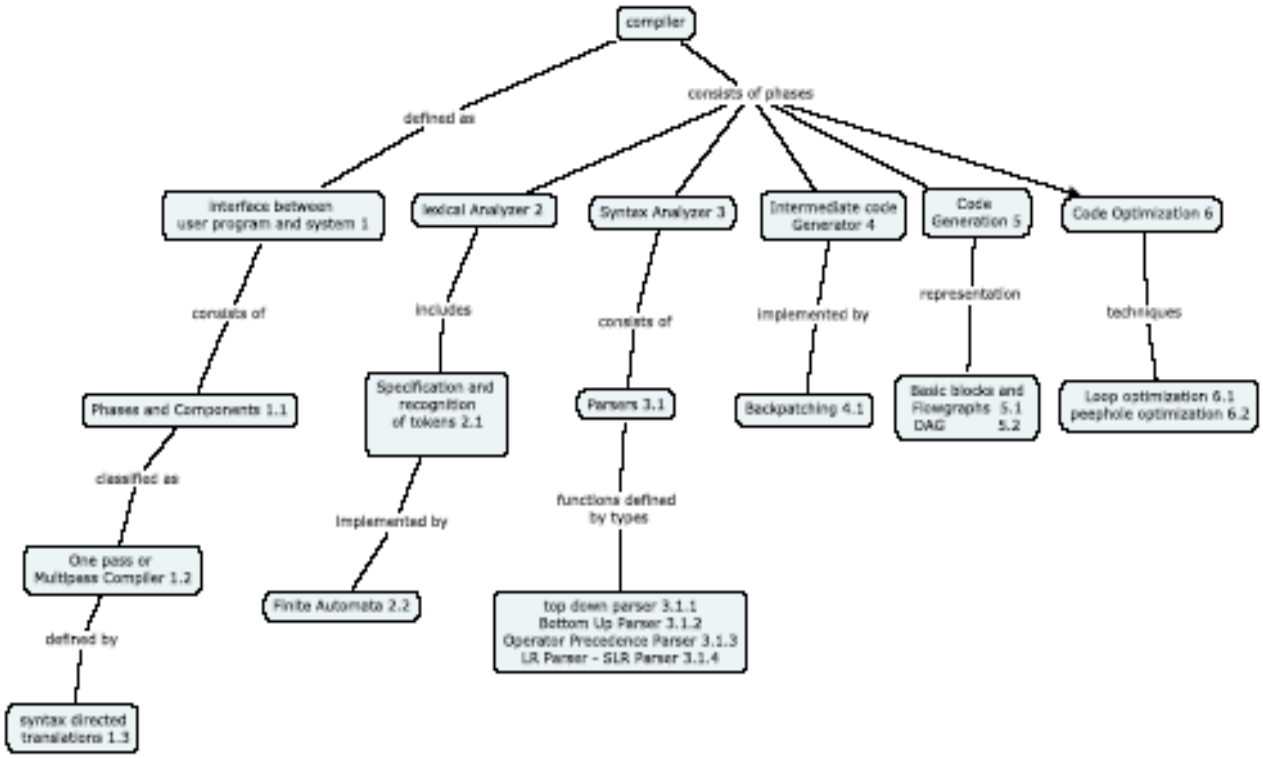
   E->E+T/T

   T->T*F/F

   F->(E)/id
8. Construct the predictive parser for the following grammar.
   S->a|↑|(T)
   T->T,S|S
   Write down the necessary algorithms and define FIRST and FOLLOW.
   Show the behaviour of the parser in the sentences:
   (i)(a,(a,a))
   (ii)(((a,a), ↑,(a),a).

9. Check whether the following grammar is a LL(1) grammar
   S->iEtS| iEtSeS' |a
   E->b

Concept Map



Course contents and Lecture schedule

S .no	Topics	No.of periods
1.	**Introduction to Compiling**	
1.1	Phases and components of a compiler	3
1.2	One pass compiler	3
1.3	Syntax-directed translation	2
2.	**Lexical Analysis**	
2.1	Specification and recognition of tokens	4
2.2	Finite Automata	4
3.	**Syntax Analysis**	
3.1	Parsers	1

3.1.1	Top-down Parsing	3
3.1.2	Bottom-up parsing	1
3.1.3	Operator-precedence parsing	2
3.1.4	LR - SLR parser	2
4.	**Intermediate Code Generation**	
4.1	Backpatching	6
5.	**Code Generation**	
5.1	Basic blocks and flow graphs	2
5.2	The DAG representation	3
6	**Introduction to Code Optimisation**	
6.1	Loop optimization	3
6.2	Peephole Optimisation	1
	Total	40

Syllabus

Introduction to compiling: Compilers, Analysis of the source program, Phases of a compiler, Cousins of a compiler, Grouping of Phases, Compiler – construction tools. **A Simple one pass compiler:** Overview, syntax definition, syntax-directed translation, Parsing, A translator for simple expressions. **Lexical Analysis:** The role of the lexical analyzer, Input buffering, specification of tokens , recognition of tokens ,A language for specifying lexical analyzers, Finite Automata. **Syntax Analysis :** The role of a parser, Context-free grammars, Writing a grammar, Top-down Parsing, Bottom-up parsing, Operator-precedence parsing , SLR parser. **Intermediate Code Generation:** Backpatching **Code Generation:** Issues in the design of a code generator, basic blocks and flow graphs, The DAG representation of basic blocks, Code generation from DAG's. **Introduction to Code Optimization:** The principal sources of optimization, Peephole Optimization, Loops in flow graphs

Text Books:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D Ullman – Compiler Principles, Techniques and Tools, Pearson Education, 2007.

Reference Books:

1. Steven S. Muchnick: Advanced Compiler Design & Implementation – Harcourt Asia, Morgan Kaufmann,2001.
2. J. P. Bennet: Introduction to Compiling Techniques, Tata McGraw-Hill Publishing-2002.

Course Designers

1. **B. Thanga Parvathi btpcse@tce.edu**
2. **M.P. Ramkumar mpr@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
C 44	3	1	-	4

C44 Operating Systems: Principles and Design 3:1

Preamble: This course is offered in the fourth semester for the students of Computer Science and Engineering. As prerequisites, the courses such as C34: Operating Systems: Configuration and use, C23: Computer Organization and Design, C25: Data Structures, and C24: Computers and Programming has been offered in the previous semesters. A course on 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. Based on this, the 'Operating Systems: Principles and Design' course is offered as a successor of C34. 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.

Program Outcomes addressed

a. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, and mathematics and science problems.
b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.
c. Graduate will demonstrate an ability to engage in life-long learning.

Competencies

1. Explain what operating systems are, what they do and how they are designed and constructed.
2. Describe the process concept, process scheduling, interprocess communication, process synchronization, threads and deadlock handling.
3. Describe the various approaches to memory management while several processes coexist in memory.
4. Explain how the file system, mass storage and I/O are handled in a modern computer system.
5. Discuss the design issues of 2 modern, most popular, drastically different operating systems, GNU/Linux and Windows.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	20	20
2	Understand	30	30	20
3	Apply	40	50	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. What is an operating system?
2. What is a time sharing system?
3. Give several definitions of a process
4. What is critical section?
5. What is storage compaction?
6. What is a) Demand paging b) Pure demand paging?
7. What is rotational latency?
8. List out the file access methods.
9. Explain inode structure in Linux.
10. What are the file access permissions in Linux?

Understand

1. What are the three major activities of an operating system in regard to memory management?
2. What is the purpose of system calls?
3. In which of the following operations, the scheduler is not called into play?
    a. Process requests for I/O.
    b. Process finishes execution.
    c. Process finishes its time allotted.
    d. All of the above through c
    e. None of the options a through c above.
4. List three examples of deadlock that are not related to a computer system.
5. Why are segmentation and paging sometimes combined into one scheme?
6. Why page sizes are always power of 2?

7. How is the information organized along sectors on a disk?
8. Explain the concept of buffering? How is the double buffering scheme organized?
9. How does Kernel help resolve system calls?
10. The NT VM manager uses a two-stage process to allocate memory. Why is this approach beneficial?

Apply

1. Using system calls, write a program in either C or C++ that reads data from one file and copies it to another file.
2. 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. What is the turnaround time for process p3?
   c. What is the average wait time for the processes?
3. Give a solution to the readers-writers problem after explaining its nature?
4. 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.
5. 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.
6. 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.

7. 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
8. 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?
9. 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.

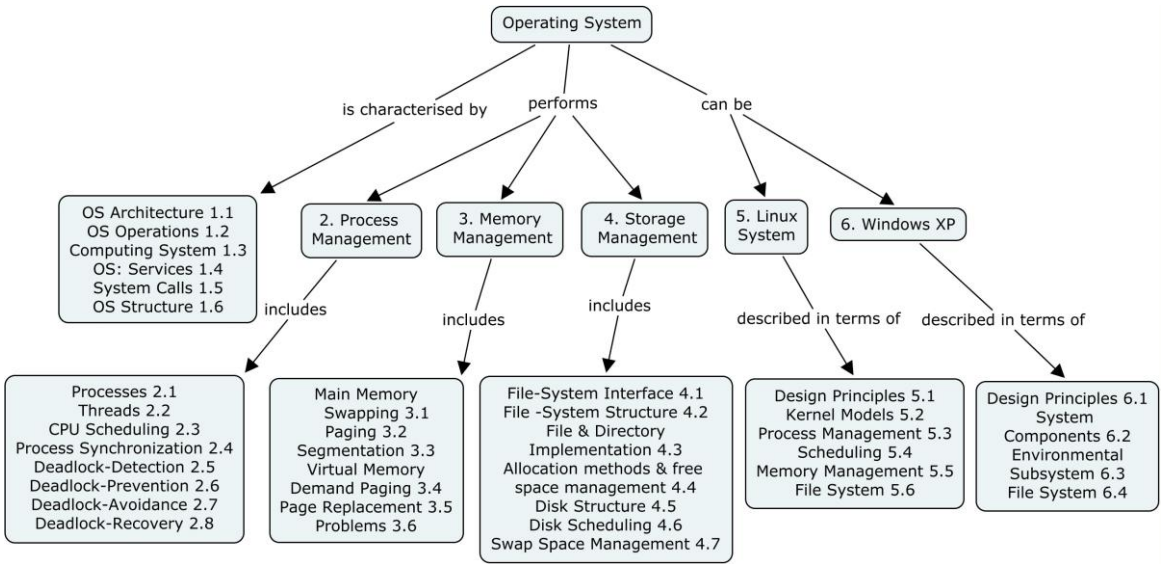
10. Consider a demand-paging system with the following time-measured utilizations

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 prepaging to the page fetch algorithms.
h. Increase the page size.

Concept Map:



Syllabus:

Operating System Introduction & Structure: Basics, OS Architecture, OS Operations, Computing Environments and systems, OS Services, System calls, OS Structure. **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, Problems. **Storage Management:** File System Interface, File System structure, File and Directory Implementation, Allocation methods, free space management, Disk Structure, Disk Scheduling, Swap-Space Management. **Case Studies: Linux System** – Design Principles, Kernel Models, Process Management, Scheduling, Memory Management, File system. **Windows XP -** Design Principles, System Components, Environmental subsystem, File system.

Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Operating System Introduction & Structure**	
1.1	OS Architecture	2
1.2	OS Operations	1
1.3	Computing Environments and systems	1
1.4	OS Services	1
1.5	System calls	1
1.6	OS Structure	1
2	**Process Management**	
2.1	Processes	1
2.2	Threads	1
2.3	CPU Scheduling	2
2.4	Process synchronization	2
2.5	Deadlock Detection	1
2.6	Deadlock Prevention	1
2.7	Deadlock Avoidance	1
2.8	Deadlock Recovery	1
3	**Memory Management**	
3.1	Main Memory – swapping	1
3.2	Paging	1
3.3	Segmentation	1
3.4	Virtual Memory – Demand paging	1
3.5	Page Replacement	1
3.6	Problems	1
4	**Storage Management**	
4.1	File System Interface	1

4.2	File System structure	1
4.3	File and Directory Implementation	1
4.4	Allocation methods and free space management	1
4.5	Disk Structure	1
4.6	Disk Scheduling	1
4.7	Swap-Space Management	1
Case Studies		
5	**Linux**	
5.1	Linux System – Design Principles	1
5.2	Kernel Models	1
5.3	Process Management	1
5.4	Scheduling	1
5.5	Memory Management	1
5.6	File system	1
6	**Windows XP**	
6.1	Windows XP – Design Principles	1
6.2	System Components	1
6.3	Environmental Subsystem	1
6.4	File System	1
	Total No of Hours	40

Text Book:

1. Avi Silberschatz, Peter Baer Galvin and Greg Gagne: Operating System Concepts, Seventh edition, John Wiley and Sons, 2006.

Reference:

1. Andrew S. Tanenbaum, Albert S.WoodHull: Operating Systems, Design and Implementation, Third Edition, Prentice Hall, 2006.
2. William Stallings: Operating Systems: Internals and Design Principles, Fifth Edition, Prentice Hall, 2004.

Course Designers:

1. J. Jane Rubel Angelina janerubel@tce.edu
2. M. Suguna mscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 45	3	1	-	4

C45 Software Engineering **3:1**

Preamble: The main objective of this subject is to promote the practice of software engineering concepts at a higher level of abstraction, in a more engineering-like fashion.

Program me Outcomes addressed

d. Graduates will identify, formulate and solve engineering problems.

e. Graduates will use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

h. Graduates will able to work on multidisciplinary teams.

Competencies

1. Explain and compare various software life cycle development models.
2. Explain the role of decomposition techniques in project estimation.
3. Explain how Risk estimation is done during project development.
4. Explain with examples, the various methods of how black box testing is performed for any software.
5. Explain the various phases of Software Reengineering.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	10	10
2	Understand	40	40	30
3	Apply	30	30	40
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	20	20

Learning Objectives

Remember

1. Which is more important? The process or the product. Justify.
2. What are the various metrics for S/W Quality?
3. What is Formal Technical Review?
4. What are the important elements of CSR in SCM?
5. How is status reporting done after S/W changes?
6. Differentiate verification with validation.
7. What is the difference between restructuring and Forward Engineering?

8. What are different approaches to the sizing problem?
9. How are test cases generated during loop testing?
10. Quality and reliability are related concepts but are fundamentally different in a number of ways. Justify.
11. What are the various maintenance activities done in maintenance of a S/W?
12. What is the difference between off-the-shelf components and full-experience components?
13. List out the software risks?
14. What are the steps in the configuration auditing?
15. Give the checklist for use during configuration audits.

Understand

1.Explain the application where agile process model is implementing?
2.Explain the need for measurement. What are the various classifications of measurement?
3.What are the advantages and disadvantages of various S/W development life cycle models?
4.Explain how testing is performed in various levels of the software. Give examples.
5.Explain the various decomposition techniques.
6.Explain the various risk strategies.
7.Difference between change control and version control in SCM.
8.Discuss about Software reviews.
9.Explain the difference between "Known Risks and Predictable risks" and develop a risk mitigation strategy and specific risk mitigation activities for any 3 risks.
10.Explain with examples, the various methods of how Black Box testing is performed for any software?
11.Explain Empirical estimation model with suitable example.
12.Explain project scheduling and tracking.
13.There is a subtle difference between restructuring and forward engineering. What is it?
14.Why should some software metrics be kept "private"? Provide examples of three metrics that should be private. Provide examples of three metrics that should be public.
15.Who should perform the validation test—the software developer or the software user? Justify your answer.
16.Compare and contrast process and project metrics in your own words.

Apply

1. Using the Cocomo models, estimate the effort required to build a S/W for a Simple ATM that produces 24 screens, 15 reports and will require approximately 110 S/W components. Assume average complexity and average developer / environment maturity. Use the application composition model with object points.
2. Select which software life cycle model is applicable for weather monitoring system.
3. Implement the Empirical estimation model for any real time application.
4. Write a function of Make / Buy decision.
5. Write down the steps to calculate Earned Value Analysis?
6. Implement the formal technical reviews for weather monitoring S/W.
7. Explain how SCM is used?
8. Explain the various Black Box Testing Techniques.
9. Explain the overall testing strategy used for client server systems
10. Compute the 3D function point value for a project with the following information domain characteristics {45, 55, 20, 25, 6, 27, 5 }. The code designed for the project is reusable, all master files are updated on-line and the performance is critical. The other complexity adjustment values are considered to be average.
11. Draw the CFG and calculate the cyclomatic complexity for the following program.

```
int compute_gcd(x, y)
int x, y;
{
while (x! = y){
if (x>y) then
x= x – y;
else y= y – x;
 }
 return x;
}
```

12. Illustrate the cost impact of early defect detection during the software process, assuming the following error detection percentage. Preliminary design-65%, detailed desing-45%, code/unit test-70% and others-50%. Assume your own amplification factors and error generated in each phase.

Create

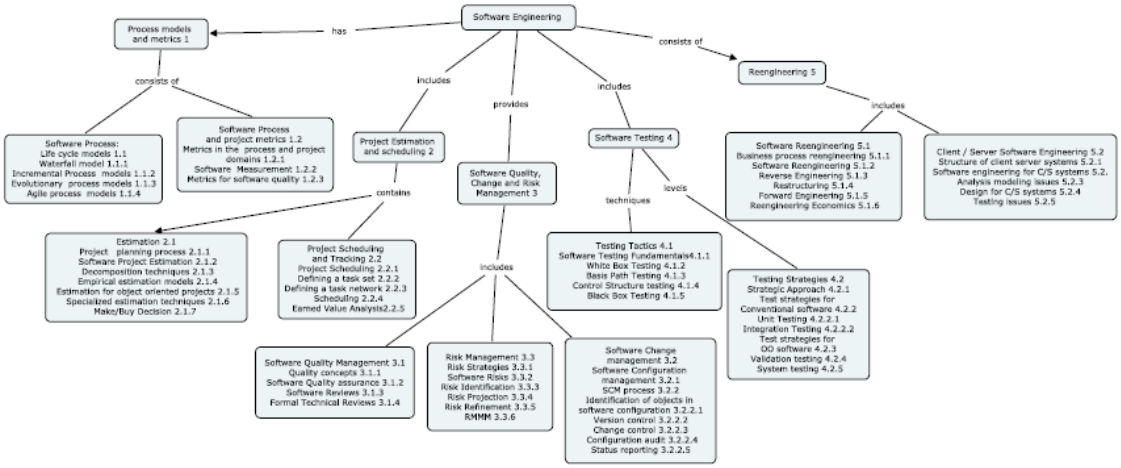
1. Design a Client-Server system for a software organization highlighting its structure.
2. Calculate the metric that finds complexity of the following program and design the test cases.

```
/*  SelectionSorter.java */

   public class SelectionSorter
   {
     public void sort(int [] A)
     {
          for (int i = A.length - 1; i > 0; i--)
          {
                int maxIndex = i;
                for (int j = i - 1; j >= 0; j--)
                {
                      if (A[maxIndex] < A[j])
                            maxIndex = j;
                }
                int temp = A[i];
                A[i] = A[maxIndex];
                A[maxIndex] = temp;
          } // for
     } // sort
   } // SelectionSorter
```

3. You're the project manager for a major software company. You've been asked to lead a team that's developing "next generation" word-processing software. Create a risk table for the project.
4. Design a project database system that could enable a software engineer to store, cross reference, trace, update, change and so forth all important configuration items. How would database handle different versions of the same program? How will two developers precluded from making different changes to same SCI at the same time?
5. 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.
6. Using the equivalence partitioning method, construct a set of test cases for above program

Concept Map



Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Process models and metrics**	
1.1	**Software Process: Life cycle models**	
1.1.1	Waterfall model	0.5
1.1.2	Incremental Process models	1.5
1.1.3	Evolutionary process models	1
1.1.4	Agile process models	1
1.2	**Software Process and project metrics**	
1.2.1	Metrics in the process and project domains	1
1.2.2	Software Measurement	2
1.2.3	Metrics for software quality	1
2.	**Project Estimation and scheduling**	
2.1	Estimation	
2.1.1	Project planning process	0.5
2.1.2	Software Project Estimation	0.5
2.1.3	Decomposition techniques: LOC based, FP based, Use-case based Estimation	2
2.1.4	Empirical estimation models	0.5

2.1.5	Estimation for object oriented projects	0.5
2.1.6	Specialized estimation techniques	0.5
2.1.7	Make/Buy Decision	0.5
2.2	Project Scheduling and Tracking	
2.2.1	Project Scheduling	0.5
2.2.2	Defining a task set	0.5
2.2.3	Defining a task network	0.5
2.2.4	Scheduling	1
2.2.5	Earned Value Analysis	0.5
3.	**Software Quality, Change and Risk Management**	
3.1	Software Quality Management	2
3.2	Software Change management	3
3.3	Risk Management	3
4	**Software Testing**	
4.1	Testing Tactics	
4.1.1	Software Testing Fundamentals	0.5
4.1.2	White Box Testing	0.5
4.1.3	Basis Path Testing	1.5
4.1.4	Control Structure testing	1
4.1.5	Black Box Testing	1
4.2	Testing Strategies	
4.2.1	Strategic Approach	0.5
4.2.2	Test strategies for Conventional software	1
4.2.3	Test strategies for OO software	1
4.2.4	Validation testing	0.5
4.2.5	System testing	0.5
5	**Reengineering**	

5.1	Software Reengineering	
5.1.1	Business process reengineering	1
5.1.2	Software Reengineering	1
5.1.3	Reverse Engineering	1
5.1.4	Restructuring	1
5.1.5	Forward Engineering	1
5.1.6	Reengineering Economics	1
5.2	Client / Server Software Engineering	2
	Total	**40**

Syllabus

Process models and metrics: Software Process: Life cycle models, Waterfall model, Incremental Process models, Evolutionary process models, Agile process models. Software Process and project metrics: Metrics in the process and project domains, Software Measurement, Metrics for software quality

Project Estimation and scheduling : Estimation: Project planning process, Software Project Estimation, Decomposition techniques, Empirical estimation models, Estimation for object oriented projects, Specialized estimation techniques, Make/Buy Decision. Project Scheduling and Tracking: Project scheduling, Defining a task set, Defining a task network, Scheduling, Earned Value Analysis

Software Quality, Change and Risk Management: Software Quality Management: Quality concepts, Software Quality assurance, Software Reviews, Formal Technical Reviews. Software Change management: Software Configuration management, SCM process, Identification of objects in software configuration, Version control, Change control, Configuration audit, Status reporting. Risk Management: Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, RMMM.

Software Testing: Testing Tactics: Software Testing Fundamentals, White Box Testing, Basis Path Testing, Control Structure testing, Black Box Testing. Testing Strategies, Strategic Approach, Test strategies for Conventional software, Unit Testing, Integration Testing, Test strategies for OO software, Validation testing, System testing.

Reengineering: Software Reengineering: Business process reengineering, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering, Reengineering Economics. Client / Server Software Engineering: Structure of client server systems,

Software engineering for C/S systems, Analysis modeling issues, Design for C/S systems, Testing issues

Text Books :

1.Roger S. Pressman, Software Engineering A Practitioner's Approach, sixth Edition, McGraw Hill International Edition.,2007

Reference Books:

1.Roger S. Pressman, Software Engineering A Practitioner's Approach, Fifth Edition, McGraw Hill International Edition.,2000

2. Ian Somerville-Software Engineering, John Wiley and sons,2003

3. Pankaj Jalote: An Integrated Approach to Software Engineering - Narosa Publishers - 1994.

4. Stephen R. Schach: Classical and Object Oriented Software Engineering, Irwin, McGraw Hill - 1996.

Course Designers

1. A. Malini malini_vsaravanan@tce.edu
2. R. Suganya rsuganya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C46	3	1	-	4

C46 - Databases: Practice **3:1**

Preamble: This course aims at facilitating the student to understand the various functionalities of DBMS software and perform many operations related to creating, manipulating and maintaining databases for Real-world applications and introducing emerging technologies in Databases.

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

At the end of the course the student will be able to

1. Understand the structure and model of the relational database system.
2. Create and modify the database with constraints.
3. Retrieve data from Databases including selecting rows, limiting the selection, and single-row functions.
4. Display data from multiple tables, and using group functions, sub queries.
5. Create other database objects like index, sequence, views, and also to control user access.
6. Develop Stored Procedures, composite data types and to handle cursor, errors.
7. Manipulate the database through stored procedures.
8. Develop database applications using high-level languages by making use of JDBC or ODBC.
9. Analyze the database using Data mining Techniques.

Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	10	10
2	Understand	30	30	30
3	Apply	50	60	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. What is database management system?
2. What is a schema?
3. Explain the structure of a DBMS.
4. Mention the types of constraint with an example for each type.
5. Give the format of date in Oracle.
6. What is the use of having clause?
7. Mention the use of Commit and Rollback commands.
8. How will you allow other user to access your objects?
9. What is a cursor?
10. Mention the use of Exception.
11. How will you connect other high level languages to your database?
12. What is Market-Basket analysis?
13. What are the two steps in Classification?
14. What are the limitations of K-means clustering?

Understand

1. Differentiate between Union and Intersect operation.
2. Compare the Unique and Primary keys.
3. List the difference between the Truncate and Drop commands.
4. How will you list out the records from a table without redundancy?
5. Will the null values be counted by the count function?
6. When the inserted or updated records will be physically stored in the disk?
7. How can constraints be created and enforced on views?
8. Give the advantage of %rowtype.
9. When to use Triggers?
10. Give the difference in accessing your table by PL/SQL and JAVA.
11. Distinguish Classification from Prediction.
12. How do you evaluate the accuracy of a Classifier?

Apply

1. 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)

Construct the appropriate tables along with the required constraints.

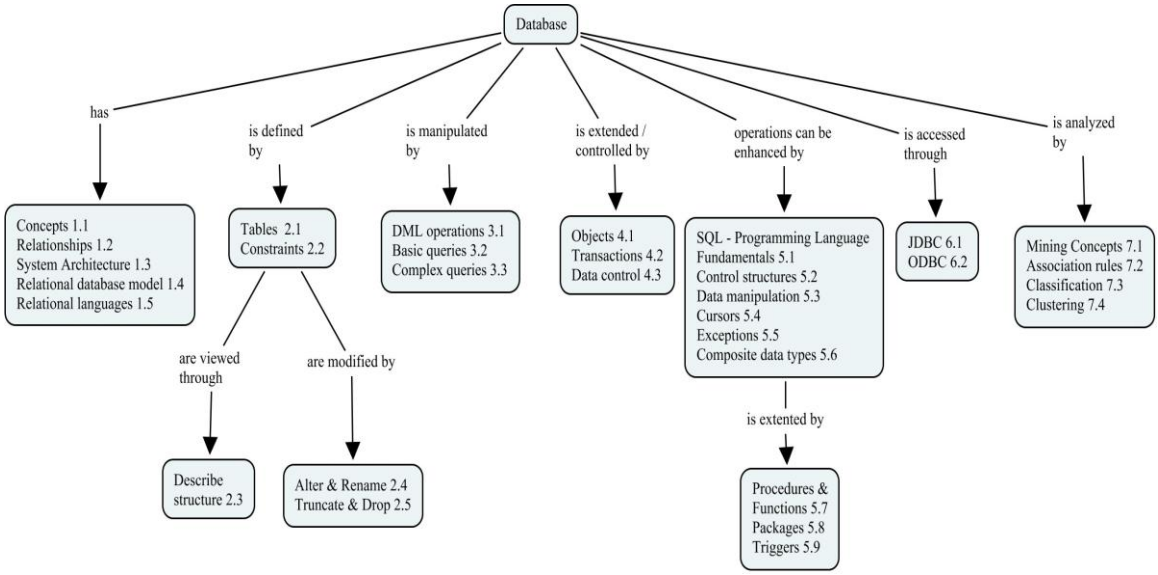
2. If you want to remove the primary constraints that you have created for the employee table, how will you do that?
3. If you want to change the city name of the company "TCS", what will you do?
4. Find those companies whose employees earn a higher salary, on average than the average salary at TCS.
5. Delete all the employees in TCS who earn less than Rs.10,000 .
6. If you want to get back the employees in TCS who earn less than Rs.10,000, what will you do?
7. You allow the user "A" to access the information regarding the employee's address along with the company in which they are working.
8. Create a procedure to display the employee's company details along with their manager information.
9. Restrict the number of employees under each manager to three. How will do this?
10. Write a Java program to list the name of the employees whose living city is same as that of their company's city.
11. For the given database find all the frequent item sets using Apriori algorithm with min_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*}

12. Classify the following Customer Database using ID3 algorithm.

Rid	**Age**	**Income**	**Student**	**Credit Rating**	**Class**
1	<= 30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3.	31…40	High	No	Fair	Yes
4.	>40	High	No	Fair	Yes
5.	>40	Medium	No	Fair	Yes
6.	31.. 40	Low	Yes	Excellent	No
7.	<=30	Low	Yes	Excellent	No
8.	<=30	Medium	No	Fair	Yes
9	<=30	Low	Yes	Excellent	No
10.	<= 30	High	No	Fair	No

Concept Map



Course Contents

No.	Topic	No. of Lectures
1	**Introduction**	
1.1	Concepts	1
1.2	Relationships	1
1.3	System Architecture	1
1.4	Relational database model	1
1.5	Relational languages	1
2.	**Data Definition Language**	
2.1	Tables creation	1
2.2	Constraints	1
2.3	Displaying table information	1
2.4	Altering and Renaming an existing table	1
2.5	Truncating and dropping the table	1
3	**Data management and Retrieval**	
3.1	DML operations – Insert , Delete & Update	1
3.2	Basic queries – select	0.5

No.	Topic	No. of Lectures
3.2.1	Arithmetic operations , where clause , Sorting	0.5
3.2.2	CASE structure , DEFINE command	1
3.3	Complex queries – Functions and Grouping	1
3.3.1	Multiple tables : JOINs and SET operators	2
3.3.3	Sub queries : Nested queries	2
4	**Advanced Features**	
4.1	Objects – Views , Sequences , Synonyms , Index	2
4.2	Transactions	0.5
4.3	Data control	0.5
5	**Programming Language Extension to SQL**	
5.1	Fundamentals	1
5.2	Control structures	1
5.3	Data manipulation	1
5.4	Cursors	2
5.5	Exceptions	1
5.6	Composite data types – Records , Tables & Varrays	1
5.7	Procedures & Functions	1
5.8	Packages	1
5.9	Triggers	2
6	**Database Connectivity**	
6.1	ODBC	2
6.2	JDBC	2
7	**Mining Techniques**	
7.1	Fundamentals	1
7.2	Association rules	1

No.	Topic	No. of Lectures
7.3	Classification	1
7.4	Clustering	1

Syllabus

Introduction - Concepts, Relationships, System Architecture, Relational database model, Relational languages **Data Definition Language:** Tables creation, Constraints, Displaying table information, Altering and Renaming an existing table, Truncating and dropping the table. **Data management and Retrieval** - DML operations – Insert, Delete & Update, Basic queries – select, Arithmetic operations, where clause, Sorting, CASE structure, DEFINE command, Complex queries – Functions and Grouping. **Advanced Features -** Objects – Views, Sequences, Synonyms, Index, Transactions **Programming Language Extension to SQL -** Fundamentals, Control structures, Data manipulation, Cursors, Exceptions, Composite data types – Records, Tables & Varrays, Procedures & Functions, Packages, and Triggers. **Database Connectivity:** ODBC, JDBC. **Mining Techniques:** Fundamentals, Association rules, Classification, Clustering.

Text Books:

1. Silberschatz, Korth, Sudarshan: "Database System Concepts", 5th Edition, Tata McGrawHill, 2006.
2. Nilesh Sha: "Database Systems using Oracle", 2nd edition, Prentice Hall of India Pvt Ltd, 2007.
3. Arun K. Pujari: "Data Mining Techniques", Universities Press, 2001.

Reference Books:

1. Joseph C. Trezzo, "Oracle PL/SQL (Tips and Techniques) ", TMH.
2. Sharad Maheshwari, Ruchin Jain: "Database Management System- Complete practical Approach", Firewall Media, Second Edition, 2006.
3. Ian H Witten and Eibe Frank: Practical Machine learning tools and techniques with java implementation", Morgan Kaufmann publications, 2000

COURSE DESIGNERS:

1. A.M.Rajeswari amrcse@tce.edu
2. B.Subbulakshmi bscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 47	-	-	1	1

C47 Network Programming Lab **0:1**

Preamble: The global connectivity can be achieved through computer networks. It is important to understand the function of computer networks and the knowledge about hardware and software requirements of networks is essential. This course emphasizes the various components to make a network operational.

LIST OF EXPERIMENTS

1. Write a program to obtain an IP Address of a given/any host.
2. Write a program to implement a time server.
3. Write a program to illustrate a simple client/server communication.
4. Write a program to implement remote method invocation.
5. Write a program to implement ECHO and PING commands.
6. Write a program to implement a file transfer using TCP/IP.
7. Write a program to find which port is currently used/ scanning the port.
8. Configuration of various networks components - connections, BNC, RJ-45, I/O box, Cables, Co-axial, twisted pair, UTP, NIC (network interface card), Switch, hub.
9. Establishment of a LAN/ Preparing of network cables
10. Use of protocols in establishing LAN
11. Installation of network device drivers
12. Installation of networks (Peer to Peer Networking client server interconnection)

Course Designers:

1. S. Srinivasan **ssnit@tce.edu**
2. S. Muthuramalingam **smrit@tce.edu**
3. S. Gilbert Nancy **gilbertnancyl@tce.edu**
4. P.Subathra **pscse@tce.edu**

| Sub Code | Lectures | Tutorial | Practical | Credit |

C48	-	-	1	1

C48 Data Bases Lab 0:1

Preamble:

This Lab aims at giving adequate exposure to the SQL and programming language extension to SQL within the RDBMS environment and usage of data mining tool.

LIST OF EXPERIMENTS:

1. Creation and Modification of Tables using Oracle and DB2.
2. Integrity Constraint enforcement using Oracle.
3. Simple SQL Queries in DB2.
4. Complex SQL Queries in Oracle.
5. Creation and usage of other database objects in Oracle.
6. Declaration and use of Cursors in Oracle.
7. Creation of Procedures and functions using Oracle and DB2.
8. Creation of Package and Triggers in Oracle.
9. Creation of composite data types in PL/SQL.
10. Database application using JDBC.
11. Database application using ODBC.
12. Association Rule Mining, Classification and Clustering using WEKA.

COURSE DESIGNERS:

1. A.M.Rajeswari amrcse@tce.edu
2. B.Subbulakshmi bscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C49	1	-	1	2

C49 PROFESSIONAL COMMUNICATION 1:1

(Common to ALL branches of B.E)

Subject codes: B49, T49,C49, D49, E49, G49, IT49

Preamble: Professional communication aims to develop Listening, Speaking, Reading and Writing skills in Engineering students' professional development contexts such as projects, competitive exams, organizational communication and soft skills.

Competencies: At the end of the course the students should be able to

Listening:

1. Listen and understand the project presentations, competitive exam exercises, organizational communication activities
2. Listen to the lectures on soft skills for practice.

Speaking:

1. Present project reports, self introduction and SWOT analysis.
2. Participate in GD , interview and team work with etiquette.

Reading:

1. Read and collect information for project report writing.
2. Read and understand the comprehension passages given in competitive examinations.
3. Read and understand the company profile

Writing:

1. Write a project report adhering to proper format
2. Create a paragraph and essay using their own ideas
3. Write circulars, minutes of the meetings, and curriculum vitae

Assessment Pattern:

	Internal (50)	**External (100)**
1. Recall	10	10
2. Understanding	10	20
3. Application	10	20
4. Analysis	10	30
5. Evaluation	5	10
6. Creation	5	10

Course Content:

1. Listening:

1.1 Attending, understanding , responding and asking questions.

1.2 Project report writing, competitive exam exercises, organizational communication and soft skills practice

2. Speaking:

2.1 Planning, preparation and presentation

2.2 Project report, self introduction, SWOT analysis, GD, interview, Team work and etiquette

3. Reading:

3.1 Rapid reading and reference skills

3.2 Project reports, competitive exam exercises and company profiles

4. Writing:

4.1. Structure

4. 1.1 Sentence structure

4.1.2 Abstract writing

4.1.3. CV writing

4.1.4. Project report writing

4.2 Organizational Communication

4.2.1 Circulars

4.2.2 Minutes of the meeting

Syllabus:

Listening: Listening to Project presentation: Asking Questions, Listening test as conducted in TOEFL and BEC, Listening in the context of Organizational communication and Soft skill practice.

Speaking: Project presentation skill, Speaking in the context of Group Discussion, Interview, , TOEFL and BEC Exam Spoken Test, Speaking in the work Contexts : Self introduction, SWOT analysis Team work, Etiquette, Mini Presentation.

Reading: Reference Skills for Project Report Writing: Topic selection, Data Collection. Rapid Reading, Reading comprehension tests conducted in CAT, TOEFL, GRE and BEC, Reading skills in work situation: Company Profile.

Writing: Project Report Writing : Format, Abstract, Bibliography, Structure : Sentence structure, CV Writing, Writing in Work context : Circulars, Minutes of the meeting.

References

1. Tony Lynch: Study Listening. Cambridge, Cambridge University Press, 2007
2. Sangeeta Sharma and Binod Mishra: Communication Skills for Engineers and Scientists. New Delhi, PHI Learning Pvt. Ltd. 2009.

3. Hari Mohan Prasad and Uma Rani Sinha: Objective English for Competitive Examination. New Delhi, Tata McGraw – Hill, 2005
4. Bob Dignen, Steeve Flinders et. al.: Work and Life: English 365. Students Book 1,2 & 3. New Delhi, Cambridge, 2004.

List of Lecture sessions:

Listening:

1. Effective listening skills
2. Nature of listening tests in competitive examinations
3. Introduction of soft skills

Speaking:

1. Introduction of Presentation skills
2. Suggestions for speaking tests in competitive exams
3. How to participate in GD
4. Interview techniques

Reading:

1. Rapid reading techniques
2. Reference skills
3. Suggestions for reading tests in competitive exams

Writing:

1. Format of project report
2. Abstract of the project
3. Sentence structure
4. Organizational communication like sending circulars, writing minutes of the meetings
5. CV writing

List of Practice Sessions:

Listening:

1 Messages, descriptions, conversations and lectures

Speaking:

1. Self Introduction, SWOT analysis and Etiquette.
2. Mini Presentation
3. GD ,Team Work
4. Interview

Reading:

1. Rapid reading practices
2. Comprehension exercises
3. Topic selection and data collection for project report

Writing:

1. Sentence structure
2. Abstract writing
3. Project Report Writing
4. Circulars
5. Minutes of the meeting
6. Model test

Course Designers

1. T. Sadasivan sadasivan@tce.edu
2. S. Rajaram sreng@tcee.edu
3. A Tamil Selvi tamilselvi@tce.edu

40th meeting of the Academic Council Resolution No. 40 :13

Annexure-W

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

FIFTH SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2000 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

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

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the year 2013-2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

Board of studies Meeting on 24.04.2010 Approved in 40th Academic Council meeting on 05.06.2010

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014 onwards)

FIFTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
C 51	Statistics And Graph Theory	BS	4	-	-	4
C 52	Databases Principles And Design	DC	3	1	-	4
C 53	Multicore architectures	DC	3	1	-	4
C 54	Computer Networks: Principles	DC	3	-	-	3
C 55	Web Programming	DC	3	-	-	3
C 56	Software Design	DC	3	1	-	4
PRACTICAL						
C 57	Web Programming Lab	DC	-	-	3	1
C 58	Software Design Lab	DC	-	-	3	1
Total			**19**	**3**	**6**	**24**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
DC : Department Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS

(For the candidates admitted from 2013 - 2014 onwards)

FIFTH SEMESTER

S.No.	Sub. code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C 51	Statistics And Graph Theory	3	50	50	100	25	50
2	C 52	Databases Principles And Design	3	50	50	100	25	50
3	C 53	Multicore architectures	3	50	50	100	25	50
4	C 54	Computer Networks: Principles	3	50	50	100	25	50
5	C 55	Web Programming	3	50	50	100	25	50
6	C 56	Software Design	3	50	50	100	25	50
PRACTICAL								
7	C 57	Web Programming Lab	3	50	50	100	25	50
8	C 58	Software Design Lab	3	50	50	100	25	50

* Continuous Assessment evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub code	Lectures	Tutorial	Practical	Credit
C51	4	-	_	4

C51 STATISTICS AND GRAPH THEORY 4:0

Preamble: An engineering student needs to have some basic mathematical tools and techniques. 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 the theory and applications of Statistics, Probability, Sampling and Graph Theory.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of Mathematics, Science and Engineering.

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

c. Graduates will develop confidence for self education and ability for life-long learning.

Competencies

At the end of the course the student should be able to

1. Find the correlation between the variables (linear, non-linear)for predictions and engineering applications.
2. Express the probability distributions arising in the study of Engineering problems and their applications.
3. Construct the various tests essentially needed for testing of samples for testing for different attributes.
4. Understand the basic concepts of graph theory used in computers application.
5. Construct the minimal spanning tree by using various algorithms.

Assessment Pattern

	Bloom's category	Test 1	Test 2	Test 3 / End Semester Examinations
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level learning objectives

Remember

1. Distinguish between
2. Define multiple correlation and Non linear regression.
3. Define Discrete and Continuous Random Variables
4. State the properties of probability distribution function.
5. Interpret the mean and variance of Gamma distribution.
6. What do you mean by test of Hypothesis?
7. Write down the applications of F-test
8. Define connected and disconnected graph.
9. How many numbers of edges in a complete graph K_n?
10. How many numbers of pendent vertices are in a binary tree?

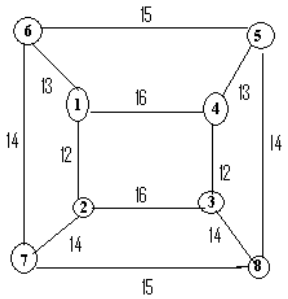
Understand

1. Discuss the uses of t – distribution.
2. Discuss the properties of a Normal probability distribution
3. Discuss the procedure for testing of hypothesis.
4. Interpret the level of significance and degrees of freedom for a Chi-square variant.
5. Discuss the properties of tree.
6. Evaluate the centre for P_n.
7. What is the total number of edges in a k-regular graph of order n ?

Apply

1. A coin is tossed 900 times and heads appeared 490 times. Would you conclude that the coin is a biased one?
2. Two random samples gave the following

   $n_1 = 10; \sum (x_i - \bar{x})^2 = 90$; $n_2 = 12$; $\sum (y_i - \bar{y})^2 = 108$. Test whether the samples came from the same population.
3. Obtain a minimum spanning tree in the graph using prim's algorithm



4. Describe the Prism's algorithm.
5. Write down any five properties of incidence matrix

Course content and lecture schedule

No	Topic	No. of Lectures
1	**Statistics**	
1.1	Linear correlation and regression	3
1.2	Curve fitting – Method of least squares	3
1.3	Multiple Regression	2
1.4	Non linear regression	2
2	**Distributions**	
2.1	Discrete and Continuous distributions	1
2.2	Binomial, poisson,	3
2.3	Geometric and Hyper Geometric distributions, Uniform,	3
2.4	Exponential, Gamma ,Weibull and Normal distributions,.	3
3	**Test of Hypothesis**	
3.1	Testing Hypothesis involving means and proportions	1
3.2	small samples t-test	2
3.3	F test	2
3.4	large samples- z-test	3
3.5	Test of goodness of fit, Chi square test	2
4	**Graph Theory**	
4.1	Basic definitions in graphs,walk,path,circuits	3
4.2	Connected and disconnected, components	2
4.3	Euler graphs, operations on graph	3
4.4	Hamiltonian graphs	2
5	**Trees**	
5.1	Properties of trees, Distance and centers in a tree	3
5.2	Rooted and binary trees, spanning trees	3
5.3	Spanning tree algorithm, kruskal's algorithm, Prism's algorithm	3
5.4	Adjacency matrix, incidence matrix.	1
	Total	50

Syllabus

Statistics: Linear correlation and regression, Curve fitting – Method of least squares, Multiple Regression, Non linear regression.

Probability distributions: Discrete and Continuous distributions- Binomial, Poisson, Geometric and Hyper Geometric distributions, Uniform, Exponential, Gamma, Weibull and Normal distributions.

Test of Hypothesis: Testing hypothesis involving means and proportions, small samples-t-test, F-test and large samples -z-test, Test of goodness of fit, Chi square test.

Graph Theory: Basic definitions in graphs, walk, path and circuits, connected graphs, disconnected graphs and components, Euler graphs, operations on graphs, Hamiltonian graphs.

Trees: Properties of trees, distance and centres in a tree, rooted and binary trees, spanning trees. Algorithm-spanning tree algorithm, Kruskal's algorithm and Prim's algorithm, adjacency matrix and incidence matrix

Text Books

1. S.C Gupta and V.K.Kapoor, "Fundamentals of mathematical statistics", Sultan Chand & Co, 2002
2. Veerarajan T., "Probability and Random Processes" TMH, 2006
3. Narsingh Deo, "Graph Theory", Prentice-Hall of India, 2004

References

1. Lecture Notes by the faculty of department of Mathematics, TCE, Madurai
2. P. Kandasamy,K.Thilagavathy, K. Gunavathy, 'Probability Random Variable and Random Processes' S.Chand & Co, 2004.
3. V.K. Balakrishnan "Theory and Problems of Graph Theory" Schaum's outlines, Tata Mc- Graw-Hill Publishing Company Limited, New Delhi,2004.

Course Designers

1. V.Mohan vmohan@tce.edu
2. G.Jothilakshmi gjlmat@tce.edu
3. A.P.Pushpalatha appmat@tce.edu
4. K.Angaleeswari kangaleeswari@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C52	3	1	_	4

C52 – DATABASES PRINCIPLES AND DESIGN **3:1**

Preamble: This course aims at facilitating the student to understand the various designing concepts, storage methods, querying and managing databases.

Program Outcomes addressed

a. Graduates will demonstrate an ability to design and conduct experiments, as well as to analyze and interpret data.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

c. Graduate will demonstrate an ability to identify, formulate and solve engineering problems.

Competencies

At the end of the course the student will be able to

1. Design a database based on a data model appropriate to the need.
2. Normalize a given database to a specified level
3. Determine the structure of the storage system and retrieval methods.
4. Estimate the storage size of the database and design appropriate storage techniques.
5. Compute the performance of a given query
6. Select the best query that meets the specified performance requirements of an application.
7. Identify the requirements of transaction processing and concurrency control.
8. Solve the concurrency control issues.
9. Perform backup and recovery of the database.
10. Design and develop a simple database application working in a team using common tools and environment.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	20	20	30
3	Apply	40	40	40
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	20	20	20

Remember

1. Define strong entity set with an example.
2. Define the term ACID properties.
3. How the database system is advantageous than file system?
4. Write a note on join dependency.
5. Outline the desirable properties of decomposition.
6. Explain the addition and deletion operations by making use of B+ tree.

Understand

1. How will you check whether the given relation can undergo 5NF?
2. Describe the join dependency with an example
3. In what way can you implement atomicity in transactions?
4. Outline the desirable properties of decomposition.
5. Stable storage can't be implemented. Explain why.
6. For the query

   select T.branch_name from branch T, branch S where T.assests > S.assests and S.branch_city='madurai'

   write an efficient relational_algebra expression for this query.

Apply

1. Consider the following transactions

T1: read(A);
read(B);
if A=0 then B:=B+1;
write(B)
	T2: read(B);
	read(A);
	if B=0 then A:=A+1;
	write(A)

   Add lock and unlock instructions to the above transactions and show that they observe two-phase locking protocol.
2. Compute the closure of the following set F of functional dependencies for relation schema R=(A,B,C,D,E).

   A->BC , CD->E, B->D, E->A

   List the candidate keys for R.
3. Using the functional dependencies given A->BC , CD->E, B->D, E->A Compute B+.
4. Let relations r1(A,B,C) and r2(C,D,E) have the following properties: r1 has 20,000 tuples, r2 has 45,000 tuples, 25 tuples of r1 fit on one block, and 30 tuples of r2 fit on one block. Estimate the number of block transfers and seeks required using Hash join strategy for r1 natural joined with r2.
5. List out all the functional dependencies satisfied by the relation. Explain how they are satisfied.

A	B	C
a1	b1	c2
a1	b1	c2
a2	b4	c1
a3	b2	c3

6. Consider the following two transactios:

T1: read(A);
read(B);
if A=0 then B:=B+1;
write(B)
	T2: read(B);
	read(A);
	if B=0 then A:=A+1;
	write(A)

   Let the consistency requirement be A=0 or B=0, with A=B=0 the initial values. Show that every serial execution involving these two transactions preserve the consistency of the database.

Create

1. 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
2. Design a database for the Banking environment by following the various design phases including normalization.
3. Construct a B+ tree for the following set of key values {
   2,3,5,7,11,17,19,23,29,31 }
4. 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.
5. Consider a relational database with two relations

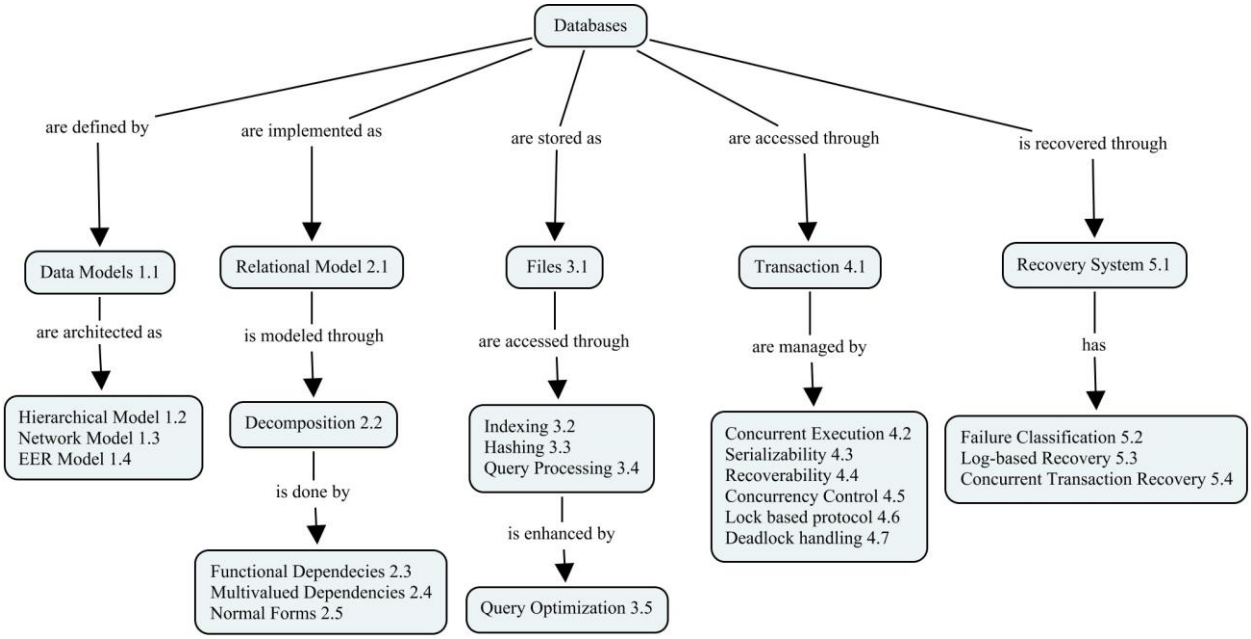
   Course(course_name,room,instructor)

   Entrollment(course_name, student_name, grade)

   Create the instances of these relations for three courses, each of which enrolls five students.
6. 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

Concept Map



Course Contents

No.	Topic	No. of Lectures
1	**Introduction**	
1.1	Data Models	1
1.2	Hierarchical Model	1
1.3	Network Model	1
1.4	EER Model	3
2.	***Database Design***	
2.1	Relational model	1
2.2	Decomposition	1
2.3	Functional dependencies	2
2.4	Multi valued dependencies	3

No.	Topic	No. of Lectures
2.5	Normal Forms	3
3.	***Storage Structure***	
3.1	File Structure	2
3.2	Indexing – Ordered Index, Bitmap index, index files, Tutorial	3
3.3	Hashing – Static and dynamic, Tutorial	3
3.4	Processing – Measures and Evaluation of query, Tutorial	2
3.5	Optimization – Estimation and Evaluation, Tutorial	3
4	**Transaction and Concurrency control**	
4.1	Transactions - concepts	1
4.2	Concurrent Execution	2
4.3	Serializability	2
4.4	Recoverability	2
4.5	Concurrency Control	2
4.6	Lock based protocol	1
4.7	Deadlock handling	2
5	**Recovery System**	
5.1	Recovery System	2
5.2	Failure Classification	2
5.3	Log-based Recovery	2

No.	Topic	No. of Lectures
5.4	Recovery with Concurrent Transaction	3
	Total	50

Syllabus

Introduction – Concepts, Data Models, Hierarchical model, Network Model, EER Model, **Database Design** – Relational Model , Decomposition, Functional Dependencies, Multivalued Dependencies, Normal forms, **Storage Structure** - File Structure, Indexing, Ordered Index, Bitmap index, index files , Hashing, Static and dynamic hashing, Processing – Measures and Evaluation of query, Optimization – Estimation and Evaluation, **Transaction and Concurrency control** - Transaction concepts, Concurrent Execution, Serializability, Recoverability, Concurrency Control, Lock based protocol, Deadlock handling, **Recovery System** - Failure Classification, Log-based Recovery, Recovery with Concurrent Transaction.

Text Books:

1. Silberschatz, Korth, Sudarshan: "Database System Concepts", 5th Edition, Tata McGrawHill, 2006.

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 DESIGNERS:

1. M.Sivakumar mskcse@tce.edu
2. A.M.Rajeswari amrcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 53	3	1	-	4

C53 MULTICORE ARCHITECTURES **3:1**

Preamble: The era of Moore's law induced performance gains of sequential programs is over. In the future, the primary method of increasing program performance will require the utilization of multicore parallelism. The processing platforms of the future will have hundreds or even thousands of processor cores that are tightly integrated for parallel processing. It is necessary to train the programmers of the future with the right concepts and tools so that they can harness the computing power of multicores and massively parallel systems. Hence, a course on Multicore architectures is an essential part of any computer-science education. This course is offered in the fifth semester for the students of Computer Science and Engineering. As prerequisites, the courses such as C23: Computer Organization and Design, C44: Operating Systems: Principles and Design C43: Principles of Compiler Design have been offered in the previous semesters.

The course will facilitate the student to understand the design trade-offs or performance of multiprocessors by knowing the interaction of applications and architecture; defining and evaluating the traditional processors and emerging multicore processors; developing parallel programs using the programming models such as shared address space and message passing; exploiting the potential processing power of multicore processors and writing efficient parallel programs; understanding the issues in the organization of shared memory multiprocessors from the system architect’s perspective

Program Outcomes addressed

1. An ability to engage in life-long learning.
2. An ability to identify, formulate and solve engineering problems.

Competencies

1. Explain what are multi-core architectures
2. Describe the issues involved in writing code for multi-core architectures
3. Convert a given sequential program into a parallel one.
4. Determine the state transitions for a given sequence of memory ‘reads’ and ‘writes’ using MSI and MESI protocol.
5. Write parallel programs using OpenMP and message passing libraries.
6. Determine the performance of a given parallel program using the performance metrics

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	20	20
2	Understand	30	30	20
3	Apply	40	50	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. What is a Multicore processor?
2. What is a shared memory multiprocessor?
3. What is cache coherence?
4. List out the different parallel programming models.
5. What are the work sharing constructs in OpenMP
6. Define the performance analysis metrics for a parallel architectures
7. Define sequential consistency.

Understand

1. What are the naming issues in the shared address and message-passing model.
2. Explain the five components of parallel execution times.
3. Compare bit level, instruction level and thread level parallelism
4. List three examples which will justify the need for memory consistency models.
5. In what way, the usage of reduction clause in openMP differs from the usage of Critical in OpenMP
6. Give the pros and cons of both message passing and shared memory programming approaches from the point of view of ease of parallel program development and performance

Apply

1. Use OpenMP directives for parallelization; write a program in C that modifies the data elements of a N X N matrix.
2. SMPs keep the private caches of the processors coherent. However, each processor in an SMP has its own private set of registers. Are they kept coherent? If not, why not? If they are not kept coherent, how is the desired semantics of a multithreaded

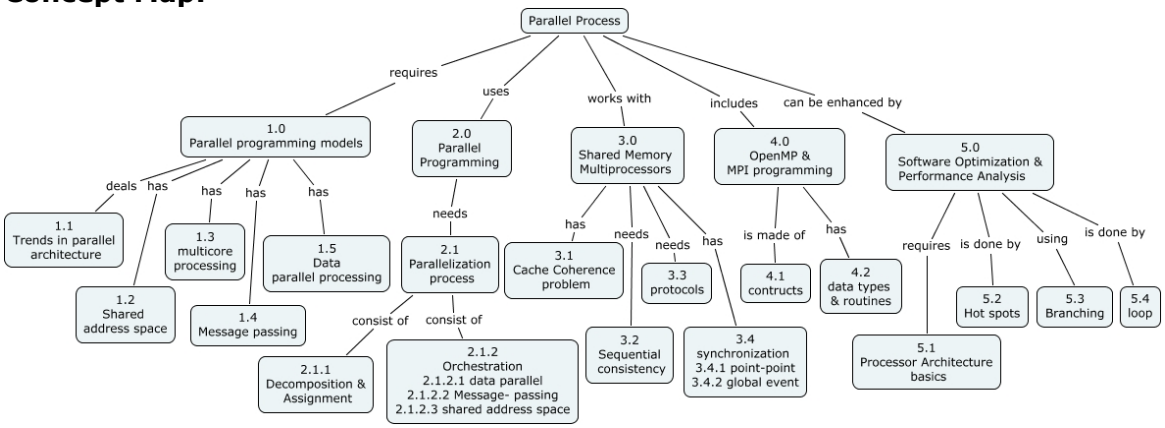
program that uses a threading package such as pthreads guaranteed to work correctly in an SMP?

3. Consider the performance of a program that exploits the "blocked" approach to multithreading to hide memory latency, where the average miss latency is 100 cycles, the average run length is 15 cycles, and the average context switch latency is 5 cycles. What is the expected processor efficiency with two threads per processor?
4. Give a solution to the cache coherence problem after explaining its nature?
5. Write a parallel program for odd-even sort and find out the performance of the program using the different performance metrics.
6. Consider the following implementation of BARRIER synchronization primitive,used at the end of each phase of the computation.Assume that bar.releasing and bar.count are initially zero and bar.lock is initially unlocked.

```
Struct bar_struct
{LOCKDEC(lock);
Int count,releasing}bar;
…….
BARRIER(N)
{LOCK(bar.lock);bar.count++;
If(bar.count ==N){
Bar.releasing=1;
Bar.count--;
{else{
UNLOCK(bar.lock);
While(!bar.releasing);
LOCK(bar.lock);bar.count--;
if (bar.count==0){
bar.releasing=0;
}
{UNLOCK(bar.lock);}
```

This code fails to provide correct barrier.Find out the problem with this implementation and correct it.

Concept Map:



Course contents and Lecture Schedule:

	Lecture Topics	**Lecture Periods**
1.0	**Parallel programming models**	
1.1	Introduction to trends in parallel architecture	2
1.2	Shared address space	2
1.3	Introduction to multicore processing	1
1.4	Message passing	2
1.5	Data parallel processing	1
2.0	**Parallel Programming**	
2.1	The parallelization process, Parallelizing computation versus data	3
2.1.1	Decomposition and Assignment	1
2.1.2	Orchestration	2
2.1.2.1	Orchestration under data parallel model	2
2.1.2.2	Orchestration under shared address space model	2
2.1.2.3	Orchestration under Message-passing model	2
3.0	**Shared Memory Multiprocessors**	
3.1	Cache Coherence problem, bus snooping	2
3.2	Sequential consistency	1
3.3	Protocols- MSI and MESI	4
3.4	synchronization event	1
3.4.1	Point-to-point synchronization	1
3.4.2	Global event synchronization	1
4.0	**OpenMP and MPI programming**	

4.1	Constructs –Parallel, worksharing and Synchronization constructs	4
4.2	MPI data types and tags , environment management routines	4
4.3	Programming exercise	2
5.0	**Software Optimization and Performance Analysis**	
5.1	Processor Architecture basics and need for optimization	4
5.2	Hot spots	2
5.3	Branching	2
5.4	Loops	2
	TOTAL	50

Syllabus

Parallel programming models- Introduction to trends in parallel architecture, Shared address space, Introduction to multicore processing, Message passing, Data parallel processing. **Parallel Programming -**The parallelization process, Parallelizing computation versus data**,** Decomposition and Assignment, Orchestration- data parallel model, shared address space model, Message-passing model**. Shared Memory Multiprocessors -** Cache Coherence problem, bus snooping**.** Sequential consistency. Protocols- MSI and MESI. synchronization event- Point-to-point synchronization, Global event synchronization. **OpenMP and MPI programming -** Constructs –Parallel, worksharing and Synchronization constructs. MPI data types and tags, environment management routines. **Software Optimization and Performance Analysis-** Processor Architecture basics and need for optimization**,** Hotspots**,** Branching, Loops

Text Book:

1. David E. Culler,Jaswinder Pal Singh, Anoop Gupta : Parallel Computer Architecture: Hardware /software approach, Elsevier Science (reprint Technical Science & Engineering) , 2008
2. Richard Gerber, Aart J.C. Bik, Kevin B. Smith, et Xinmin Tian : Software Optimization Cookbook, Intel Press, 2006

Reference:

1. Maurice Herlihy, Nir Shavit :The Art of Multiprocessor Programming, Morgan Kauffman publishers, 2008

2. Rohit Chandra, Ramesh Menon , Leo Dagum, David Kohr , Dror Maydan , Jeff McDonald: Parallel Programming in OpenMP, Morgan Kauffman publishers, 2001

Course Designers:

1. P.Chitra pccse@tce.edu
2. S.Sudha ssj@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C54	3	-	-	3

C54 COMPUTER NETWORKS: PRINCIPLES 3:0

Preamble: The course aims at exploring the principles and design of computer networks featuring the Internet, covering aspects ranging from transmitting frames on a communication link and routing packets in a network to the design of network applications. An overview of wide area technologies, wired and wireless LANs with focus on network management services is also provided.

Program Outcomes addressed

Graduates will demonstrate

a. an ability to identify, formulate and solve engineering problems.
b. an ability to design a system, component or process as per needs and specifications.
c. skills to use modern engineering tools, softwares and equipment to analyze problems.

Competencies

1. Understand the fundamentals of computer networking, including protocol design, protocol layering, algorithm design, and performance evaluation.
2. Demonstrate an understanding of design principles and specific implemented protocols covering the application layer, transport layer, network layer, and data link layer of the Internet (TCP/IP) stack.
3. Demonstrate an understanding of wide area network technologies like Frame relay and Asynchronous transfer mode; and wired and wireless local area networks.
4. Write programs for simple network applications using socket programming.
5. Determine the performance of a given network by evaluating various Quality of Service metrics.
6. Perform simulations of example networks with NS-2 and with LAN trainer kit.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	30	20	10
2	Understand	30	20	20
3	Apply	40	30	40
4	Analyze	0	20	20
5	Evaluate	0	0	0
6	Create	0	10	10

Course Level Learning Objectives

Remember:

1. Define Framing.
2. What is sub netting?
3. Define round trip time of a packet.
4. State any two real time applications where UDP datagram service is preferred.
5. Enlist the functionalities of transport layer.
6. What is a sequence number?
7. Mention any few error messages generated by Internet Control Message Protocol.
8. What if fragmentation?
9. Mention the physical devices operating at various layers of TCP/IP protocol suite.
10. What is classfull addressing?

Understand:

1. Differentiate physical address, logical address and service point address.
2. Distinguish between connection oriented and connectionless services.
3. Explain the steps involved in distance vector routing to calculate the shortest path between two nodes.
4. Explain the protocol format of a TCP header with a neat sketch.
5. Explain the steps involved in Huffman encoding.
6. Discuss the functionalities of all layers in Frame relay network.
7. Illustrate Random Early discard method of congestion avoidance with an example.
8. Draw the architecture of IEEE802.11 wireless local area network.
9. Explore the various Multiple Access techniques.
10. Compare the various flow control methods available for noisy channels.

Apply

1. Use Cyclic Redundancy check to generate the code word for the data word 1010011010 and the divisor 10111. Verify the code word at the receiver side.
2. Compute the values of k and n in the Hamming code c(n,k) with d_{min} =3 for a data word of at least 16 bits.
3. Apply one's complement addition to generate the checksum for the message "INFORMATIONTECHNOLOGY".
4. Calculate the minimum and maximum options are being carried away by this packet?
5. Compute the CLR and CER of an ATM network if it has lost 5 cells out of 12,000 and 3 are in error.
6. An organization is granted the address 211.17.180.01 /24, The administrator wants to create 32 subnets.
    a. Find the subnet mask.
    b. Find the number of addresses in each subnet.
    c. Find the first and last address in subnet 1
    d. Find the first and last address in subnet 32
7. An IPV4 packet has arrived with a few hexa decimal digits as follows

    0x 45000028000100000102……

    Compute the number of hops the packet can travel before being dropped.
8. A slotted ALOHA network transmits 200 bit frames with a shared channel with 200 kbps bandwidth. Calculate the throughput if the system produces
    a. 1000 frames/sec.
    b. 500 frames/sec.
    c. 250 frames/sec.

Analyze:

1. Does the efficiency of ATM using AAL3/4 depend on the size of the packet, assuming that there is no padding? Justify your answer.
2. Identify the changes to be made in ATM layer to accommodate an increase in data rate from 100mbps to 1gbps.
3. Examine whether a user can send data of 1 mbps all the time in a frame relay network through T-1 line, with the following specifications: CIR = 1mbps; BC = 5 million bits/5 sec and BE of 1 Million bits/5 seconds.
4. An ISP has a block of 1024 addresses. It needs to divide the addresses among 1024 customers. Does it need subnetting? Explain your answer.
5. Analyze the reasons for absence of padding in AAL1.
6. What are the effects of using CSMA, ALOHA Access control algorithms for wire less networks?

7. Can the value of the header length in IPv4 packet be less than 5? When it is exactly 5?
8. Find out the reasons for two security protocols (AH and ESP) in IPv6.
9. An IP datagram is carrying a TCP segment destined for the address 130.14.16.17/16. The destination port address is corrupted and it arrives at destination 130.14.16.19/16. How does the receiving TCP react to this error?
10. An IPv4 datagram has arrived with the following information in the header in hexadecimal 45 00 00 54 00 03 58 50 20 06 00 00 7C 4E 03 02 B4 0E 0F 02
    a. Examine the following
        i. Is the packet corrupted?
        ii. Are there any options?
        iii. Is the packet fragmented?
    b. How many routers can the packet still travel?
    c. What is the identification number and what is the type of service.

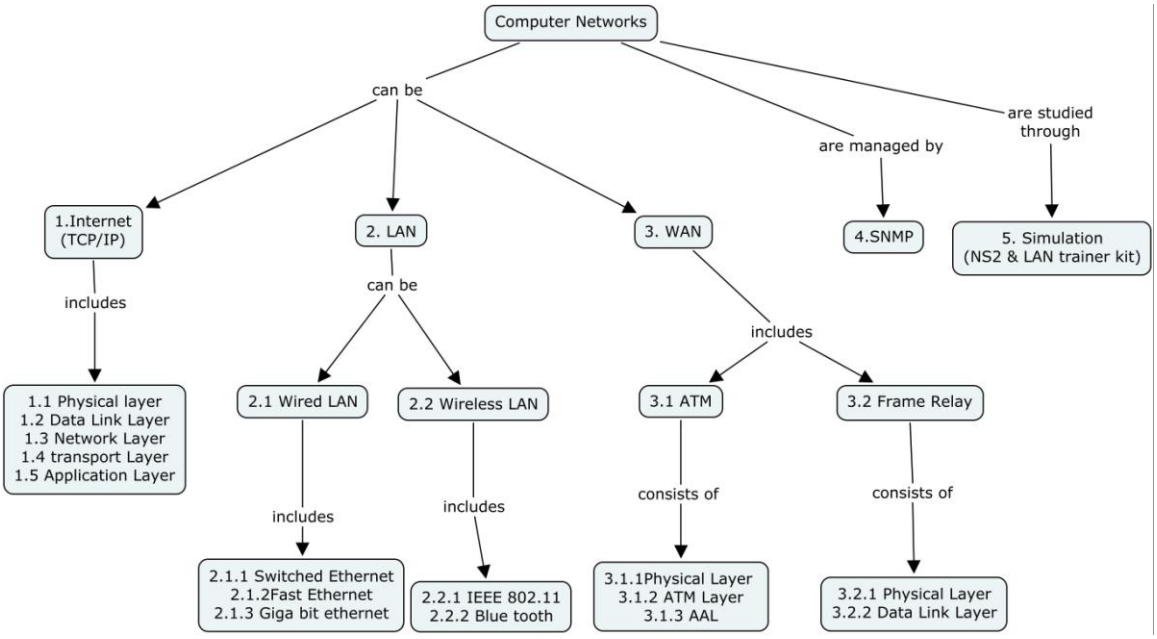
Create:

1. An ISP is granted block of addresses 190.100.0.0 /16. The ISP needs to distribute these addresses to three groups of customers as follows:
    a. The first group has 64 sub groups and each needs 256 addresses.
    b. The second group has 128 sub groups and each needs 128 addresses.
    c. The third group has 128 sub groups and each needs 64 addresses.

   Design sub blocks and find how many addresses are still available after these locations.
2. Design a bidirectional algorithm for GOBACK-N ARQ protocol using piggy backing.
3. Create a congestion window as a function of round trip time for the following scenario:

   Congestion control algorithm uses a linear increase and multiplicative decrease but not slow start. During communication packets 9,25,30,38 and 50 are lost. Assume a perfect time out mechanism that detects a lost packet exactly 1 RTT after it is transmitted.
4. Suggest your own ideas to address the limitations in sliding window Protocol.
5. Design hybrid architecture for a multimedia network using frame relay and ATM network.
6. Construct a routing table for heterogeneous networks of wired and wireless networks. Consider your own assumptions regarding the number of nodes and the configuration of network.

Concept Map



Course Contents and Lectures schedule

1.	Internet – TCP/IP model	1
1.1	Physical Layer	3
1.2 1.2.1 1.2.2	Data Link Layer Framing Physical Addressing	2
1.2.3	Flow Control	2
1.2.4	Error Control	2
1.2.5	Access control	2
1.3	Network Layer – ICMP, IGMP	3
1.3.1	Logical Addressing	2
1.3.2	Routing	2
1.4 1.4.1 1.4.2	Transport Layer Service point Addressing Segmentation and reassembly	3
1.4.3	Connection Control –TCP, UDP	2

1.4.4	Flow Control – Congestion Control, Quality of service	3
1.4.5	Error Control	1
1.5 1.5.1 1.5.2	Application Layer Dialog control Synchronization	1
1.5.3	Data Compression	1
1.5.4	Network Security	1
2 2.1 2.1.1	Local Area Networks Wired LAN switched Ethernet	2
2.1.2 2.1.3	Fast Ethernet Giga bit Ethernet	2
2.2 2.2.1	Wireless LAN IEEE 802.11	2
2.2.2	Blue tooth	2
3 3.1	Wide Area Networks Asynchronous transfer mode	2
3.2	Frame Relay	2
4	Simple Network Management Protocol	3
5	Network simulation using NS2 and LAN trainer kit	3
	TOTAL	50

Syllabus:

Types of Computer Networks: Local Area Network, Wide Area Network, Internet, Protocols and standards. **Internet Model: TCP/IP Layers Physical Layer Data link Layer –** Framing, Physical Addressing, Flow control-noisy and noiseless channels, Error Control –Error detection and Error correction codes, Access control-ALOHA,CSMA,CSMA/CD,TDMA, FDMA,CDMA. **Network Layer –** Logical Addressing – IPv4, IPv6, ICMP, IGMP, Routing – Distance Vector, Link state algorithms. **Transport Layer** – Service point addressing, segmentation and reassembly, Connection control -

TCP, UDP, Flow control-TCP congestion control, Congestion avoidance mechanisms, Quality of service, Error control. **Application Layer** – Dialog control, Synchronization, Translation – presentation Formatting, Data Compression, Network security. **Local Area Networks: Wired LAN** –switched Ethernet, Fast Ethernet, Giga bit Ethernet **Wireless LAN** – IEEE 802.11, Blue tooth. **Wide area Technologies:** Asynchronous transfer mode –Physical layer, ATM layer, ATM adaptation layer. Frame relay- Physical Layer, Data link layer. **Network Management** –SNMP, **Network simulation** using NS2 and LAN trainer kit.

Textbook

1. Behrouz A.Foruzan, "Data Communication and Networking", Tata McGraw Hill, Fourth Edition, 2009.

Reference Books

1. Larry L.Peterson and Bruce S. Davie, "Computer Networks – A systems Approach" Fourth Edition, Morgan Kaufmann Publishers, 2007.
2. Andrew.S.Tanenbaum, "Computer Networks", Prentice Hall , Fourth Edition ,2009
3. William Stallings, "Data and Computer Communications", Pearson Education, Sixth Edition.

Course Designers:

1. S.Muthuramalingam smrit@tce.edu
2. C.Jeyamala jeyamala@tce.edu
3. M.P Ramkumar mpr@tce.edu
4. T.Manikandan tmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C55	3	-	-	3

C55 WEB PROGRAMMING **3:0**

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 and protocols. Students gain understanding of how the Internet application works and develop programming skills for developing Rich Internet Applications. This course will establish a professional, client-based attitude towards web-design. Students will communicate effectively using today's technologies.

Program me Outcomes addressed

a. Graduates will use middleware wherever necessary to integrate data sources.

b. Graduates will able to use vendor support for XML.

c. Graduates will learn how to do client side as well as server-side programming.

d. Graduates will create and develop interactive web pages on the World Wide Web.

Competencies

1. Understand the architectural models of an enterprise system.
2. Understand how the Internet works with HTML, XML, Web services.
3. Understand how Internet application works in general.
4. Construct XML Documents.
5. Use XSL for transformations and formatting.
6. Use XML schemas and DTDs to construct data structure, constraints and type enforcement.
7. Develop server side programming for web application development
8. Develop programming skills in Rich Internet Applications like AJAX, Flex.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	60	20	10
3	Apply	20	60	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	20

Course Level Learning Objectives:

Remember:

1. Explain the advantages of hybrid 2 tier architecture?
2. What are the String Object XHTML markup methods?
3. What is a SOAP message?
4. What are XML Parsers?
5. List ways that Web 2.0 can be used to make you money.
6. Compare the differences between IIS/Apache and Windows/Linux in regards to web development.
7. List out JSF Components.

Understand:

1. 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?
2. Write a Java Script to get input from the user in XHTML and convert to lowercase and uppercase?
3. How to include element dimensions in style sheets?
4. Write a DTD for the XML Document formed from the above table?
5. How to display the information about the Web Browser?
6. Write a simple test program using PHP on the local server (http://localhost/)
7. How to create a Full-scale Ajax-Enabled Application?

Apply:

1. Draw the XML Schema for Book Publisher along with List of Books.
2. Write an XML and DTD for Storing Recipes.
3. Construct an XML Doc that marks up the information in the following table (10)

Name	Job	Department	Cubicle
Joe	Programmer	Engineering	5E
Erin	Designer	Marketing	9M
Melisa	Designer	Human Resource	8H
Craig	Admin	Engineering	4E
Eileen	Proj Coordinator	Marketing	3M
Danielle	Programmer	Engineering	12E
Frank	SalesPerson	Marketing	17M
Corinne	Programmer	TechSupport	19T

4. Implement the Remote Interface by using an Server side program and client side program containing the Remote Objects definition.
5. Construct an XML schema for the book's XML document
6. Construct the book's XML document using cascading style sheets.
7. Draw a picture describing the relationship between client/server objects used by PHP and mySQL

Create:

1. Create an XHTML document that marks up your resume? (10)

    a. The first document will establish the **frameset**, which will be split into at least **three** frames. One of these frame windows must contain a document that provides navigation through this set of documents (a table of contents).
    b. The rest of the set will contain, at a minimum, **six** other documents, **four** of which must pass through an active frame. These may include documents that you have created for other assignments, or other pages you have created, but if, you choose to do this, you are responsible for the markup in these pages.
    c. One document in this set must contain a working **form** with at least **three** different types of **input elements**.

    It will also include:

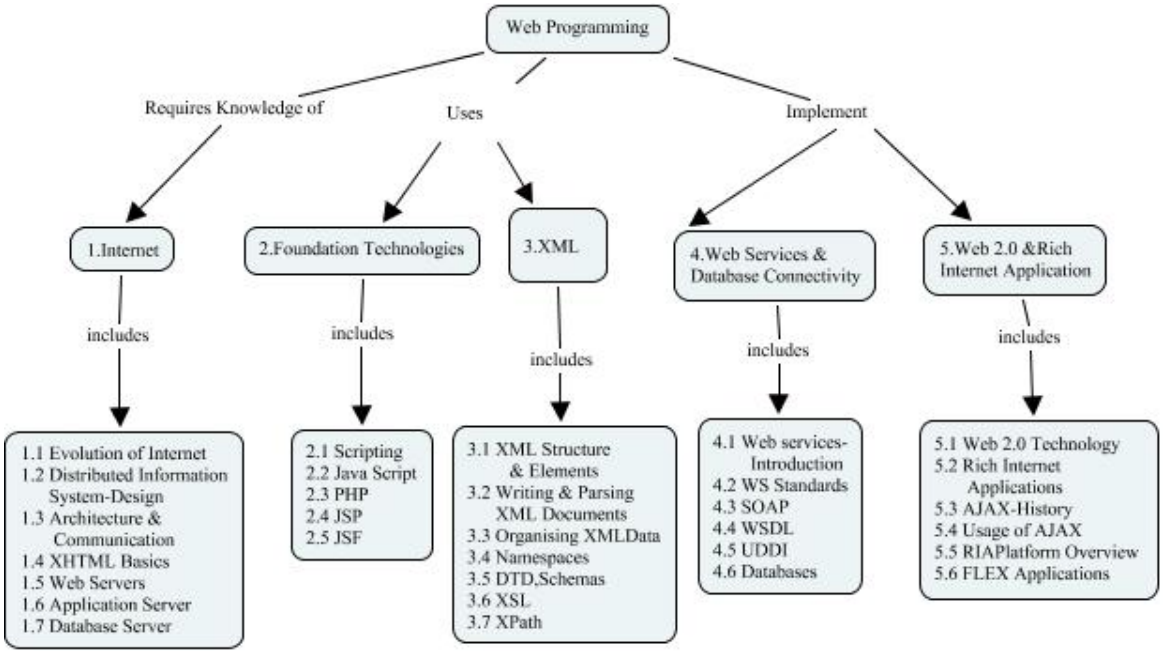
    d. At least **two** external links to documents you do not own;
    e. At least **two** links to other documents you have created;
    f. At least **one** background color or image;
    g. At least **three** images in the body of your resume pages; and
    h. At least **two** targeted links between frames
2. Create an XHTML document titled " How to get Good Grades" Use <meta> tags to include a series of keywords that describe your document?

```
<?xml version = "1.0"?>
<!DOCTYPE html  PUBLIC "/W3XX//DTD XHTML 1.1//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1.dtd>
<html xmlns = "http://w3.org/1999/xhtml">
<head>
<title> How to get Good Grades"</title>
<meta name = "keywords" content = " Way to read, study, Understanding,
concentrate, seek advice, group study"/>
</head>
<body>
<p> concentrate and seek advice to clarify the doubts</p>
</body>
</html>
```

3. Create a Web Page that applies the invert filter to all image if the user moves the mouse over it?
4. Create an XHTML document that marks up your resume?
5. Create an XHTML document titled "How to get Good Grades" Use <meta> tags to include a series of keywords that describe your document.

6. Use PHP sessions, Create persistent variables that can be used by multiple Requests from a client browser session.
7. Create a Working environment for PHP Web page development.

Concept Map



Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Architecture models (10)**	
1.1	History-Evolution of Internet, Distributed Information Systems-Design	1
1.2	Architecture of an Information system, Communication in an Information system	2
1.3	XHTML-Basics, Headers, Linking	1
1.4	Images, Image map, meta elements	1
1.4	Frameset, cascading style sheet	2
1.5	HTTP-Limitations of HTTP, Web servers-IIS & Apache	2
1.6	Application servers	1
2.	**Foundation Technologies (12)**	
2.1	Introduction to scripting, control statement,	2

	function, objects	
2.2	Java script	2
2.3	PHP-Basics, String processing and Regular Expressions	2
2.4	Connecting to a Database, Using cookies, Dynamic content	2
2.5	JSP-Java Web Technologies, creating and Running a simple application in Netbeans	2
2.6	JSF Components, Session Tracking	2
3.	**XML Basics (10)**	
3.1	XML Structure, Elements	1
3.2	writing and Parsing XML Document	1
3.3	Organizing XML data, adding attributes	2
3.4	XML Namespaces, Working with DTD, Schemas	3
3.5	Elements,types,groups,defining attributes	1
3.6	XSL,XPATH	2
4	**Web services and Database Connectivity (8)**	
4.1	Need of WS,WS Standards	1
4.2	SOAP-Goals, Structure and contents of SOAP Messages	2
4.3	Binding SOAP Message	1
4.4	WSDL-Goals, Structure	1
4.5	UDDI-Goals, Registry, Data structures	1
4.6	Databases: RDBMS, MySQL , SQL	2
5	**Web 2.0 and Rich Internet Application (10)**	
5.1	Web 2.0-Introduction,Technology overview, Integrating with web services	2
5.2	Rich Internet Application-Basics, Key characteristics	1
5.3	AJAX –History, Examples of usage of AJAX	1
5.4	Creating AJAX enabled application	2

5.5	RIA Platform overview (Flex,Silverlight)	1
5.6	Creating and customizing a simple User interface with Flex	2
5.7	Accessing XML data, Creating charts and graphs	1
	Total	**50**

Syllabus

Internet: History-Evolution of Internet, Distributed Information Systems-Design, Architecture, communication in an Information System, XHTML- headers, Linking, Images, Image map, meta elements, frameset, cascading style sheet. HTTP-Limitations of HTTP, Web servers-IIS & Apache, Application servers

Foundation Technologies: Scripting, Java script, control statement, function, objects. Server side programming, PHP-Basics, String processing and Regular Expressions, Connecting to a Database, Using cookies, Dynamic content, JSP-Java Web Technologies, creating and Running a simple application in Netbeans, JSF Components, Session Tracking.

XML : XML Structure, Elements, writing and Parsing XML Document, Organizing XML data, adding attributes, Namespaces, Working with DTD, Schemas, Elements, types, groups, defining attributes, XSL, XPath.

Web services (WS) and Database Connectivity: Need of WS, WS Standards, SOAP-Goals, Structure and contents of SOAP Message, Binding, WSDL-Goals, Structure, UDDI-Goals, Registry, Data Structures, Databases: RDBMS Model, SQL, MySQL.

Web 2.0 and Rich Internet Application: Web 2.0-Introduction,Technology overview, Integration with web services, Rich Internet Application-Basics, Key characteristics, AJAX- History, Examples of usage of AJAX, Creating AJAX-Enabled Application, RIA Platform overview(Flex,Silverlight), Creating and customizing a simple User interface, Accessing XML Data, creating charts and graphs.

Text Books:

1. Paul J.Deitel and Harvey M.Deitel, "AJAX, Rich Internet Applications, and Web Development for Programmers", Pearson Education, First Edition, 2009.

2. Deitel and Deitel, "Internet and World Wide Web How to Program", Prentice Hall of India, Fourth Edition, 2009

Reference Book:

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. James Talbot, "Total Training for Adobe Flex 3: Rich Internet & AIR applications", Adobe Incorporations.
4. http://oreilly.com/web2/archive/what-is-web-20.html

Course Designers:

1. S.Padmavathi spmcse@tce.edu
2. K.Narasima Mallikarjunan arjunkumbaraj@tce.edu
3. S.Sridevi sridevi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C 56	3	1	-	4

C56 SOFTWARE DESIGN **3:1**

Preamble: Software design is concerned with techniques, strategies, representations, and patterns used to determine how to implement a component or a system.

Prerequisite: C45

Program me Outcomes addressed

d. Graduates will identify, formulate and solve engineering problems.

e. Graduates will use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

h. Graduates will able to work on multidisciplinary teams.

Competencies

1. Explain the principles of software design
2. Understand the process of software architectural design
3. Understand the process of component level software design
4. Draw different levels of Data Flow Diagram to meet the given requirement specifications
5. Draw Jackson structured diagram to meet the given requirement specifications.
6. Create and specify the software design for a medium-size software product using Unified Modeling Language (UML).
7. Design Use Case, Class, State, Activity, Sequence, Collaboration and Deployment diagrams as per given system requirement specifications
8. Compare and contrast object-oriented analysis and design with structured analysis and design.
9. Design software systems for Web applications including e-library, air line reservation, e-banking and materials management.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	20
2	Understand	20	20	20
3	Apply	40	30	20
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	20	30	40

Learning Objectives

Remember

1. What makes software design different from coding?
2. How are the concepts of coupling and software portability related?
3. List out the guidelines to be followed in UID.
4. Differentiate the sequence and collaboration diagrams
5. Name three other generic attributes that differentiate Web Applications for more conventional software applications?
6. Suggest three "golden rules" that would help guide the design of Web Applications.
7. What are the benefits of OOAD?
8. What is relationship? Give the relationships associated with the classes.
9. Mention the importance of model building.

Understand

1. Explain SSA/SD process in detail.
2. What is a Pattern? In what way it is used for modeling? Explain the modeling techniques that are used to model the Design Pattern and Architectural Pattern of an application.
3. What are the various building blocks of UML? Explain them in detail
4. Explain the relationships among the objects with an example
5. How the dynamic nature of a system can be known? Explain the UML diagrams that are used to show the dynamic nature of the system.
6. What is dependency? Where can it be used? Explain the stereotypes that can be used along with it.
7. Explain the following:
   (i) States (ii) transition (iii) Sub state
8. What is the purpose of a component diagram? Explain the modeling techniques that is used to model the source code, executable release, physical database, and adaptable system with suitable example.
9. Discuss the following diagrams with suitable example.
   (i) Component (ii) Deployment (iii) Collaboration

Apply

1. For the lending library, what operations of a library records and issues system will need to be modeled by using:
   a. interaction function processes
   b. Information function processes
2. In a simple banking system, one candidate for the role of an 'object' might be a customer account. Consider how each of the four major viewpoints might be used in modeling the operation of a bank account, and suggest forms that might be suitable

for each viewpoint. (There is scope for using inheritance here, too, in that the bank may provide different forms of account, and although each will provide certain standard methods, the details of these might vary with the type of account involved.)

3. An issue records system for use in a public lending library is required. Suggest a set of suitable candidates for 'objects' in this system and for each of these identify the major attributes and provided operations.
4. Apply a "stepwise refinement approach" to develop three different levels of procedural abstraction for one or more of the following programs:
    a. Develop a check writer that, given a numeric dollar amount, will print the amount in words normally required on a check.
    b. Iteratively solve for the roots of a transcendental equation.
    c. Develop a simple round-robin scheduling algorithm for an operating system
5. Discuss the relative merits and difficulties of applying data flow-oriented design in the following areas: (a) embedded microprocessor applications, (b) engineering/scientific analysis, (c) computer graphics, (d) operating system design, (e) business applications, (f) database management system design
6. How does content analysis differ from interaction and functional analysis?

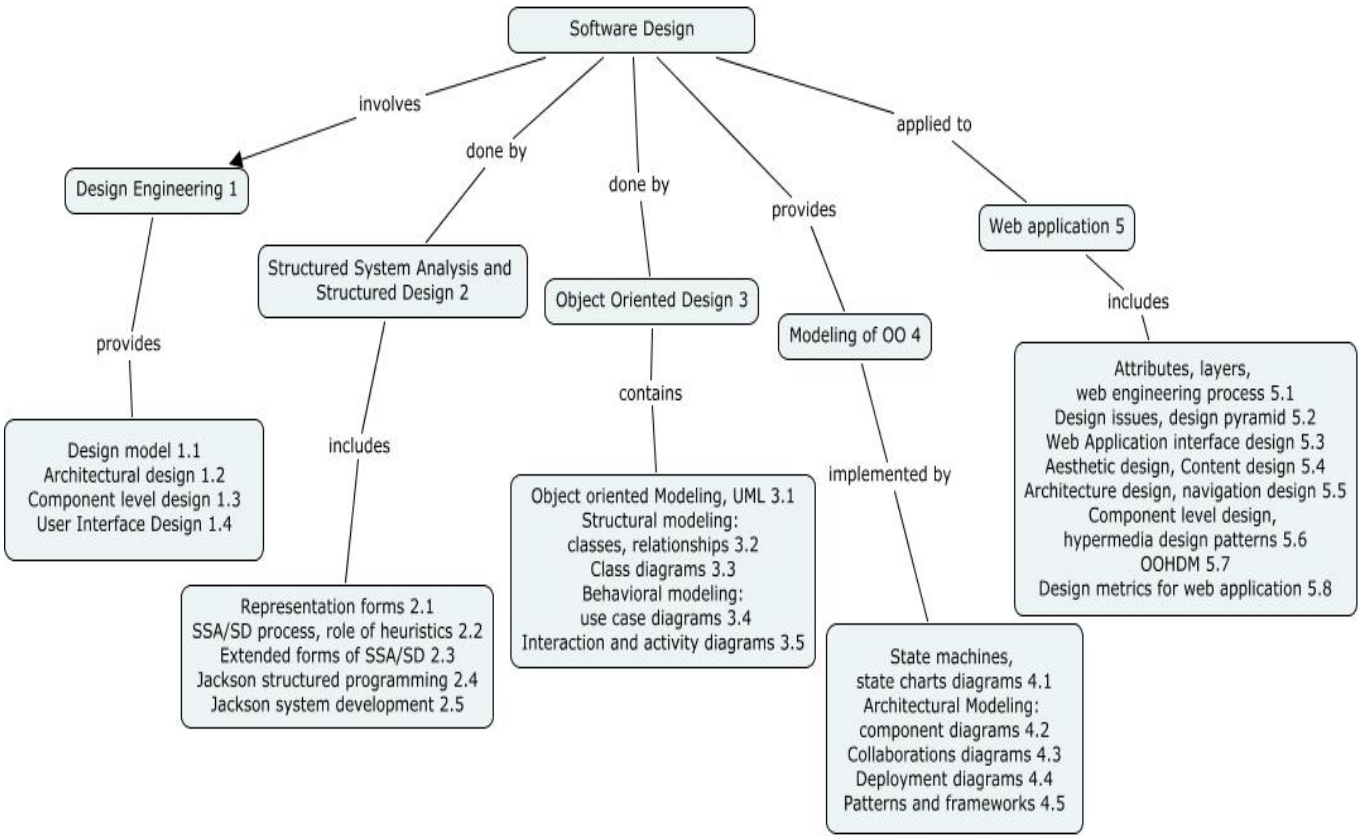
Create

1. Draw the context diagram for each of the following systems:
    (a) a bank auto teller machine
    (b) a word-processing program
    (c) a payroll package that is required to produce a set of printed pay cheques each month and to provide pay statements showing amount paid and deductions
2. Draw a Structure Diagram that describes the address of a person in the 'standard American' format of surname, forename, initials, number, street, city, state, zip code, elaborating on the details of each of these in turn. How would this model cope with Jim Smith, who lives in Blacksmith's Cottage with an address that has no street name?
3. Construct an ESD that describes:
    (a) the 'customer' of a public lending library;
    (b) the operational cycle of an automatic washing machine;
    (c) the 'customer' of a car-hire firm.

4. Develop a set of screen layouts with a definition of major and minor menu items for the advanced *Safe Home* system

5. Develop an approach that would automatically integrate error messages and a user help facility. That is, the system would automatically recognize the error type and provide a help window with suggestions for correcting it. Perform a reasonably complete software design that considers appropriate data structures and algorithms.
6. Develop a design model, a user model, a system image, and a system perception for an Internet-based polling booth for public elections.
7. For the Traffic control management system draw the Use case , Class and Object diagrams using UML.
8. For the banking environment using the UML draw the following diagrams.
   (i) Use Case (ii) Sequence (iii) Activity

Concept Map



Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Design Engineering (12)**	

1.1	Design concepts, Design model	1
1.2	Architectural design	4
1.3	Component level design	4
1.4	User Interface Design	3
2.	**Structured System Analysis and Structured Design (8)**	
2.1	Representation forms	1
2.2	SSA/SD process, role of heuristics	3
2.3	Extended forms of SSA/SD	2
2.4	Jackson structured programming	1
2.5	Jackson system development	1
3.	**Object Oriented Design (10)**	
3.1	Object oriented Modeling, UML	1
3.2	Structural modeling: classes, relationships	2
3.3	Class diagrams	2
3.4	Behavioral modeling: use case diagrams	2
3.5	Interaction and activity diagrams	3
4	**Modeling of OO (10)**	
4.1	State machines, state charts diagrams	3
4.2	Architectural Modeling: component diagrams	2
4.3	Collaborations diagrams	2
4.4	Deployment diagrams	2
4.5	Patterns and frameworks	1
5	**Web Engineering (10)**	
5.1	Attributes, layers, web engineering process	1
5.2	Design issues, design pyramid	1
5.3	Web Application interface design	1
5.4	Aesthetic design, Content design	1
5.5	Architecture design, navigation design	2

5.6	Component level design, hypermedia design patterns	1
5.7	OOHMD	2
5.8	Design metrics for web application	1
	Total	**50**

Syllabus

Design Engineering: Design concepts, Design model, Architectural design: software architecture, data design, architectural styles and patterns, architectural design, assessing alternative architectural designs, mapping data flow into a software architecture. Component level design: component, designing component-based design, conducting component level design, object constraint language, designing conventional components. User Interface Design: interface analysis, interface design steps.

Structured System Analysis and Structured Design: Representation forms, SSA/SD process, role of heuristics, extended forms of SSA/SD, Jackson structured programming, Jackson system development.

Object Oriented Design: Object oriented Modeling, UML, Structural modeling: classes, relationships, class diagrams. Behavioral modeling: use case, interaction and activity diagrams.

Modeling of OO: State machines, state charts diagrams. Architectural Modeling: component, collaborations, deployment diagrams, patterns and frameworks.

Web Engineering: Attributes, layers, web engineering process, design issues, design pyramid, Web Application interface design, aesthetic design, content design, architecture design, navigation design, component level design, hypermedia design patterns, OOHDM, design metrics for web application

Text Books :

1.Roger S. Pressman, Software Engineering A Practitioner's Approach, Sixth Edition, Mcgraw Hill International Edition.,2007

2. David Budgen "Software Design", Second Edition, Pearson Education.,2008

Reference Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson "The Unified Modeling Language – User Guide" Pearson Education, 2005

2. Ed Downs, Peter Clare, Jan Coe, "Structured System Analysis And Design Methods Application And Context", Prentice Hall, 1998.

Course Designers

1. A. Malini amcse@tce.edu
2. A.Askarunisa aacse@tce.edu

Sub code	Lecture	tutorial	Practical	credit
C57	-	-	3	1

C57 –WEB PROGRAMMING LAB 0:1

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 Rich Internet Application like Ajax, Flash and Flex and also standardization of XML Documents for the purpose of data exchange and integrate the communication mechanism also.

List of Experiments

1) Create a home page for your project using HTTP and HTML.
2) Format and Display the XML file using JavaScript.
3) Create a Webpage to handle Events and Objects using Java Script.
4) Implement Client side and Server side form validation using JavaScript
5) Write a Servlet which communicates with Browser
6) Present a XML document using cascading style sheets (CSS).
7) Implement a program for DOM to Process XML File.
8) Develop a java program to connect to database using Jdbc Drivers
   Write the Same with PHP Programming
9) Create a relational Database and display the concurred data's from two tables in a web page.
10) Create a Ajax enabled form which gets input for online Registration for paper Presentation contest (should prompt for missing Information in client side)
11) Perform object Animation using Flash Action Script.
12) Write a web program to display an flex application.
13) Create a flex Chart.
14) Create an Interactive application for server side application in flex.

Course Designers:

1)S.Padmavathi spmcse@tce.edu
2)K.Narasimha Mallikarjunan arjunkambaraj@tce.edu
3)S.Sridevi sridevi@tce.edu

Sub code	Lecture	Tutorial	Practical	credit
C58	-	-	3	1

C58 SOFTWARE DESIGN LABORATORY 0:1

Preamble: With a dynamic learn-by-doing focus, this laboratory course encourages students to explore software design methods by implementing them, a process through which students discover how software design can be applied.

Prerequisite: C56

List of Exercises:

1. Implementation of Data Flow Diagram
2. Implementation of Jackson structured programming
3. Implementation of Jackson system development
4. Implementation of Use Case Diagram
5. Implementation of Class Diagram
6. Implementation of State Diagram
7. Implementation of Activity Diagram
8. Implementation of Sequence Diagram
9. Implementation of Collaboration Diagram
10. Implementation of Deployment Diagram

A possible set of applications may be the following:

a. E-Library System
b. Inventory System
c. Course Registration System
d. Quiz System
e. Student Marks Analyzing System
f. Reservation Systems for Airlines, Railways etc.
g. Stock Management System
h. Real-Time Scheduler
i. Simulator Software for Parallel Processing Operation

Course Designers

1. A. Malini amcse@tce.edu
2.A.Askarunisa aacse@tce.edu

41st meeting of the academic council Resolution No.41 :12

Annexure-Q

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

SIXTH SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2000 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

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

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the year 2013 -2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014 onwards)

SIXTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
C 61	Management Theory & Practice	DC	3	0	-	3
C 62	Theory of Formal Languages	DC	3	0	-	3
C 63	Software Verification & Validation	DC	3	1	-	4
C 64	Cryptography	DC	4	0	-	4
CXX	ELECTIVE I	DC	3	0	-	3
CXX	ELECTIVE II	DC	3	0	-	3
PRACTICAL						
C 67	Cryptography Lab	DC	-	-	3	1
C 68	Software Testing Lab	DC	-	-	3	1
Total			**20**	**0**	**6**	**22**

BS : Basic Science
HSS : Humanities and Social Science
ES : Engineering Science
DC : Department Core

L : Lecture
T : Tutorial
P : Practical

Note:

1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SCHEME OF EXAMINATIONS

(For the candidates admitted from 2013-2014 onwards)

SIXTH SEMESTER

S.No.	Sub. code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C 61	Management Theory& Practice	3	50	50	100	25	50
2	C 62	Theory of Formal Languages	3	50	50	100	25	50
3	C 63	Software Verification & Validation	3	50	50	100	25	50
4	C 64	Cryptography	3	50	50	100	25	50
5	CXX	ELECTIVE I	3	50	50	100	25	50
6	CXX	ELECTIVE II	3	50	50	100	25	50
PRACTICAL								
7	C 67	cryptography Lab	3	50	50	100	25	50
8	C 68	Software Testing Lab	3	50	50	100	25	50

* Continuous Assessment evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub Code	Lectures	Tutorial	Practical	Credit
C61	3	-	-	3

Management Theory and Practice **3:0**

[Common to D61/ G71/ C61]

Preamble

Management is the science of managing operations for an enterprise or organization. It deals with managing men, material, machinery and money. It has become an essential need to analyze the basic concepts of management theory and to understand the ways and means of implementing them in practice. The course work highlights the systematic approach for the management of various departments in an organization .

Program outcomes addressed

b). ability to identify, formulate and solve engineering problems

g). ability to function on multidisciplinary teams

h). ability to communicate effectively in both oral and written forms

i). ability to consider social, environmental, economic and ethical impact of engineering activities in a given context

Competencies

At the end of the course, the student will be able to

1. Manage the operations in total for an enterprise.
2. Work with team spirit and group coordination.
3. Ability to design Organizational Structure
4. To facilitate an effective communication both within and outside a firm.
5. Formulate the selection and recruitment procedures for a department
6. Evolve proper performance appraisal system
7. Analyze and identify an effective site selection and design a proper layout.
8. Prepare maintenance schedules for an organization.
9. Ability to measure overall productivity and suggest means to improve it
10. Plan the material handling systems for the organization.

Assessment Pattern

SNo	Blooms Category	Tet1	Test2	Test3
1	Remember	10	10	20
2	Understand	10	10	40
3	Apply	30	30	40
4	Analyze	-	-	-
	Evaluate	-	-	-
6	Create	-	-	-

Course Level Learning Objectives

Remember

1. Define Management
2. What are the various functions of management?
3. Distinguish MBO and MBE.
4. Define Group Cohesiveness
5. What do you mean by semantic barrier of communication?
6. What type of industry requires process type layout?
7. Define Morale
8. Mention the significance of Market Research?
9. Give an example of centralized layout.
10. Mention the types of maintenance
11. What do you understand by the term Productivity?

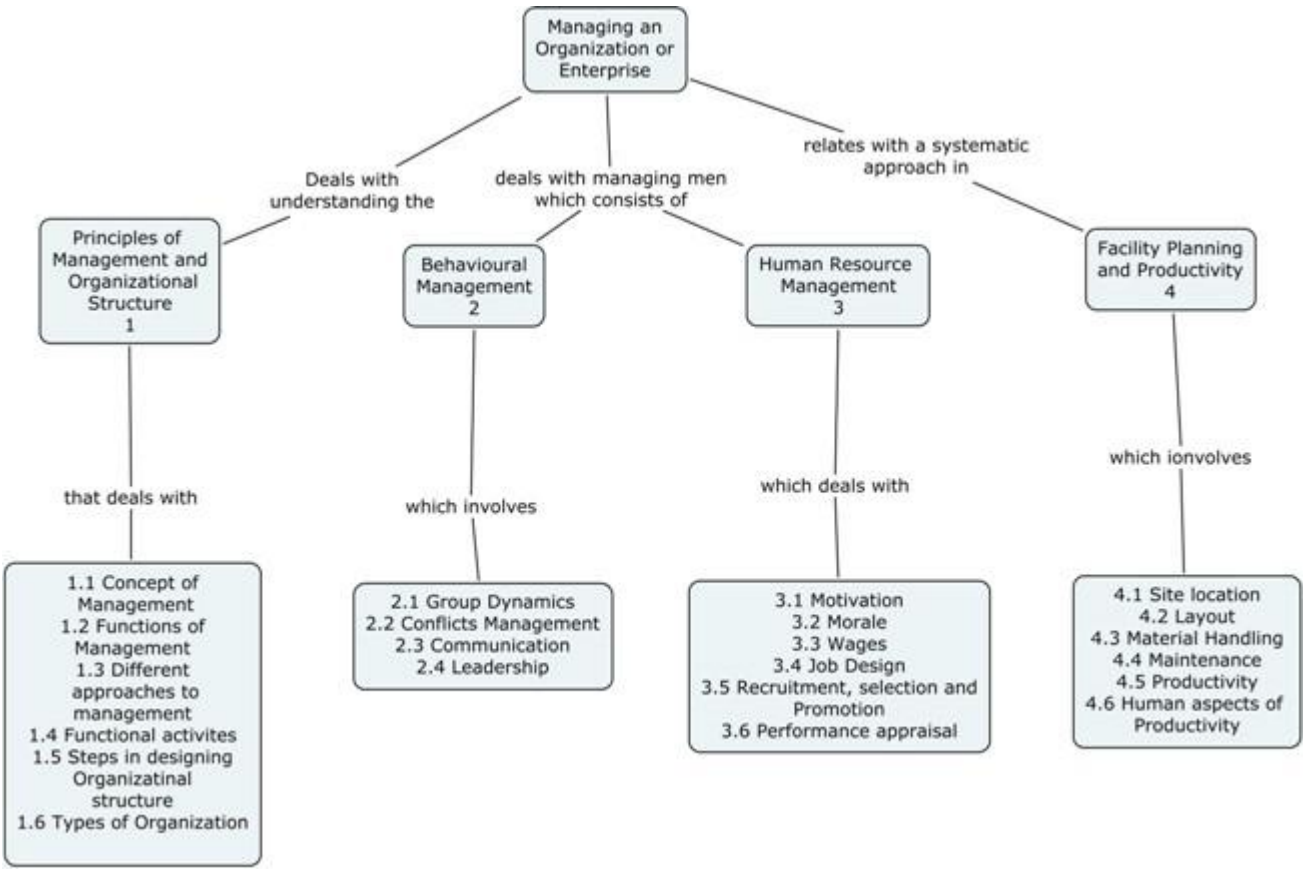
Understand

1. Briefly explain all the functions of Management
2. Explain various controlling techniques.
3. What are the merits of Modern Type of Organization?
4. List out all the stages of Group formation and explain
5. What are the barriers of communication?
6. What are the factors associated with morale?
7. Compare job enrichment and job enlargement
8. Compare the merits and demerits of product and process layouts
9. Enumerate all the human factors associated with productivity
10. Differenciate periodical and preventive maintenance
11. Enumerate all the ways of measuring productivity.
12. What are the requirements of an effective material handling system

Apply

1. Bring out all the steps in the formation of a Quality Circle in an educational institution.
2. Suggest the modalities of selection of a trainee engineer to be recruited for a software firm
3. You as a manager prepare a proposal to locate a site to establish a telecommunication industry / Automobile Industry
4. Suggest all the possible ways to increase the overall productivity of a manufacturing sector
5. Prepare a preventive maintenance schedule for an electronic equipment manufacturing company which operates for three shits in 24 hours for 8 hours per shift by 6.00 AM to 2.00 PM, 2.00 PM to 10.00 PM and 10.00 PM to 6.00 AM

Concept map



Syllabus

Management and Functions of Management

Concept of management, organization, Administration-Management is a Science or an Art- Taylors Scientific Management – Henry Fayol's Principles of management -Functions of management- planning, Organizing, Staffing, Coordinating, Directing and Controlling-different approaches to management-various functional activities of different departments-Strategic planning-Management by Objectives [MBO], Management by Exception [MBE]- Organization Structure- Principles, Steps in designing an Organization, Types of Organization.

Behavioural Management

Group dynamics, types of groups, formation of group, Group cohesiveness, conflicts management- Communication –meaning and types, barriers in communication, communication in Groups- Leadership styles

Human Resources Management

Objectives-employer-employee relations-Motivation-Morale-Ways of achieving high morale-collective bargaining-Wage and wage payments-incentives-job design ,job analysis-job description, job rotation, job evaluation and merit rating-Recruitment, Selection and training of employees-Promotion-Performance appraisal

Facility Planning and Productivity

Site location-Factors to be considered-layout-objectives, types, factors influencing layout, layout procedure-Materials handling-principles, factors affecting the choice of materials handling, Materials handling equipment-Plant maintenance-need functions and types-Productivity-definition and concept, measurement-techniques for productivity measurement-Human aspects of productivity

Text Book

1. Harold Koontz, Heinz weihrich "Essentials of Manangement", Tata McGraw Hill ,6th Edition,2004
2. O.P. Khanna, Industrial Engineering and Management , Dhanpat Rai Publications,2006

References

1. Chase, Jacobs, aquilano, "Production and Operations Managemet " 8th Editin, Tata McGraw Hil Companies Inc 1999
2. Fred Luthans "Organizational Behaviour", Tata McGraw Hill, 2005
3. Edwin Flippo, "Personnel Management", Tata McGraw Hill,2004
4. R.N. Gupta, "Principles of Management",S.Chand and

Co Ltd, 2008

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Priciples of Managmeent and Organizational Structure**	
1.1	Concept of management, Organization, Administration, Management is science or art, Taylor's Scientific Management, Henry Fayol's Priniciples of management	3
1.2	Functions of management, Planning, organizing, Staffing, Coordinating, Directing and controlling	3
1.3	Different approaches to management	1
1.4	Functional activities, Strategic Planning, MBO, MBE	1
1.5	Principles and Steps Designing Organization structure	2
1.6	Types of Organization	1
2	**Behavioural Management**	
2.1	Group Dynamics, types of group, formation of group, group cohesiveness	3
2.2	Conflicts management	2
2.3	Communication, meaning and types, barriers in communication , communication in groups	3
2.4	Leadership styles	2
3	**Human Resource Management**	
3.1	Employer employee relations, Motivation	3
3.2	Morale, ways of achieving high morale, collective bargaining	1

No.	Topic	No. of Lectures
3.3	Wages, wage and wage payments, incentives	1
3.4	Job Design, job analysis-job description, job rotation, job evaluation and merit rating	3
3.5	Recruitment, Selection and Promotion	3
3.6	Performance appraisal	1
4	**Facility Planning and Productivity**	
4.1	Site Location , factors to be considered	2
4.2	Layout objectives, types , factors influencing layout, layout procedure	2
4.3	Material Handling, principles, factors affecting the choice of materials handling, materials handling equipments	2
4.4	Maintenance, need, functions and types	2
4.5	Productivity, definition and concept, measurement-techniques for productivity measurement	2
4.6	Human aspects of Productivity	2
Total		**45**

Course Designers

1. S.Krishnan deantlp@tce.edu
2. S.Muralidharan murali@tce.edu
3. R.Muruganandham rmmech@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C62	3	-	-	3

C62 Theory of Formal Languages 3:0

Preamble: This course is offered in the Sixth semester for the students of Under Graduate Computer Science and Engineering. It introduces some fundamental concepts in automata theory and formal languages including grammar, Finite Automata, Regular expression, Formal language, Push down automata, Turing machine and Computable languages. Besides they form basic models of computation, they are also having the foundation of many branches of computer science, e.g. Compilers, software engineering, Concurrent systems etc. The properties of these models will be studied and various rigorous techniques for analyzing and comparing them will be discussed, by using both formalism and examples.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

d. Graduates will demonstrate an ability to function in multi-disciplinary terms and can participate and succeed in competitive examinations.

Competencies

At the end of the course the student will be able to

1. Explain the basic concepts of finite state and push down automata.
2. Gain knowledge of regular languages and context free languages.
3. Express the relation between regular language, context free language and corresponding recognizers.
4. Understand the concepts of Turing machine and classes of problems.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	20	20	20	20
2	Understand	40	30	30	30
3	Apply	40	20	20	20

4	Analyze	0	20	20	20
5	Evaluate	0	10	0	10
6	Create	0	0	10	0

Course Level Learning Objectives

Remember

1. Define Finite state systems.
2. List out all the types of grammars.
3. Define Turing machine.
4. State CFL?
5. Identify the applications of CFG?
6. State Push down automata?
7. State Rice theorem?
8. State the difference between Moore and Mealy machine.
9. List regular set for the following expression. 1(01)* (10)*1
10. Give formal definition of PDA.
11. State the difference between PDA and TM.
12. State Undecidability?
13. State the difference between recursive and recursively enumerable language.

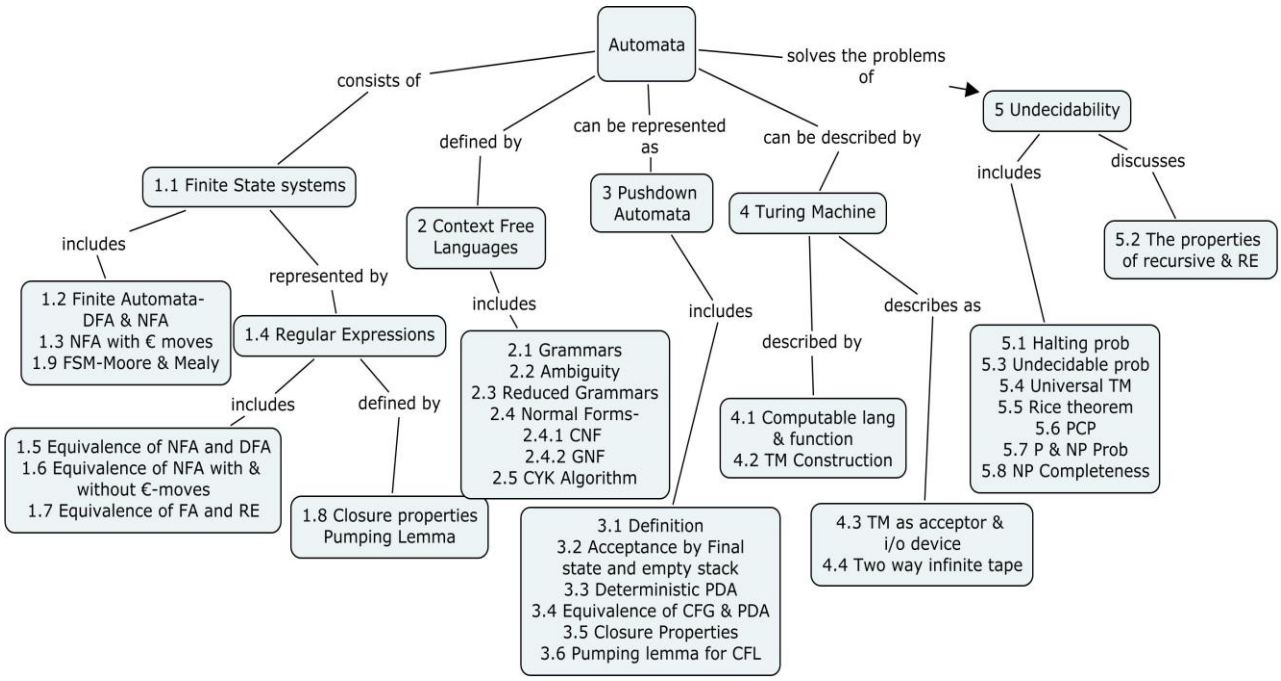
Understand

1. Describe the equivalence of NFA and DFA and prove it.
2. Explain the DFA with set of all strings such that that 10^{th} symbol from the right end is '1'.
3. Explain all Normal forms in CFG.
4. Describe about reduced grammar.
5. Illustrate finite automata that accepting {11, 110}*{0}
6. Illustrate an example of a language accepted by PDA but not by DPDA.
7. Explain any two properties of recursive language in detail.
8. Summarize a transition diagram for TM which constructs complement of a given string.
9. Explain the acceptance concepts of PDA.
10. Discuss about Chomsky hierarchy.
11. Discuss about pumping Lemma for CFL?
12. Consider G1 and G2 are any arbitrary CFG's. State that it is undecidable whether L(G1) = L(G2) is empty?
13. For a TM, carefully state what it means for M to compute the function f: $\Sigma^* \rightarrow \Sigma^*$.

Apply

1. Demonstrate the equivalence of FA and RE with an example.
2. Demonstrate the idea behind the conversion of DFA to RE.
3. Implement a PDA accepting the language L ={ equal number of X's and Y's }by empty store.
4. Demonstrate the closure properties of CFLs.
5. Construct a PDA that accepts the language generated by grammar with productions S-> aSbb | a.
6. Illustrate that the membership problem is undecidable.
7. Identify a TM M for successor function.
8. Construct a TM for a language L ={$a^n b^n$, n≥1}
9. Illustrate L_u is recursively enumerable.
10. Illustrate that PCP is undecidable.

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.

References

1. K.Krithivasan and R.Ramal; Introduction to Formal Languages, Automata Theory and Computation", Pearson Education, 2009.
2. J.E. Hopcroft and J.D Ullman, "Introduction to Automata Theory, Languages and Computation", Narosa Publishers, 2002.
3. J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Second Edition, Addison Wesley, 2001
4. H.R.Lewis and C.H. Papadimitriou, "Elements of the Theory of Computation", Second Edition, Pearson Education/ PHI, 2003.
5. J.Martin, "Introduction to Languages and Theory of Computation", Third Edition, Tata McGraw Hill, 2003.
6. R.B.Patel, Premnath "Theory of Automata and Formal Languages" First Edition, Umesh publications, 2005.
7. T.A. Sudkamp, "An Introduction to the Theory of Computer Science Languages and Machines" Third Edition, Addison Wesley, 2006.

Course content and Lecture Schedule

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
	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	2

5.4	Universal Turing machine	1
5.5	Decidability, Rice's Theorem	2
5.6	Post's correspondence problem (PCP), emptiness and equivalence problems of languages	2
5.7	Time and tape Complexity measure of TM – the classes of P and NP problems	2
5.8	NP –completeness	2
	Total	40

Course Designer:

1. Mrs. R.Suganya rsuganya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C63	3	1	-	4

C63 Software Verification & Validation 3:1

Preamble: Software verification and validation ensures that the product satisfies its specification and that the product implemented meets the expectations of the stakeholders.

Prerequisite: C45, C56

Program me Outcomes addressed

b. Graduates will identify, formulate and solve engineering problems.

e. Graduates will use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

h. Graduates will visualize and work on laboratory.

Competencies

At the end of the Course the student will be able to

1. Understand testing techniques including white-box and black-box.
2. Design test cases using test generation methods for white-box testing technique including control flow graph and data flow graph.
3. Design test cases using test generation methods for black-box testing technique including equalized partitioning, boundary value analysis, category partition, cause-effect graphing, finite state machines, W-methods, UIO sequence method and combinatorial design.
4. Determine the test adequacy criteria for all white-box and black-box testing techniques.
5. Apply mutation testing for a given program.
6. Prioritize, minimize and select test cases using methods including execution trace, and dynamic slicing.
7. Prepare test documents.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	20
2	Understand	20	20	20
3	Apply	40	40	40
4	Analyze	0	0	0
5	Evaluate	0	0	0

6	Create	20	20	20

Test 1/Test 2: 15
Test 3: 25
Assignments: 10
End-semester Examination: 50
Total: 100

Learning Objectives

Remember

1. Differentiate verification with validation.
2. How are test cases generated during loop testing?
3. What are the steps performed in Integration Testing Process.
4. What are the issues of testing a client server system.
5. What are Finite State Models?
6. What is Data Flow testing?
7. What is the difference between effort and schedule?
8. Differentiate dead and live mutant?
9. What are the properties of test data adequacy criteria?
10. What are Latin squares and mutually orthogonal Latin squares (MOLS)?

Understand

1. Explain the various Black Box Testing Techniques.
2. Explain the overall testing strategy used for client server systems.
3. Explain how testing is performed in various levels of the software. Give examples.
4. Discuss the W method for test generation.
5. Explain the various types of test case that could be generated when Condition Testing is performed. Give suitable examples.
6. How does one generate test configurations from MOLS?
7. Discuss strengths and limitations of test adequacy based on program mutation.
8. How to prioritize a set of tests for regression testing?
9. Explain how Web – based Systems are tested.
10. What is the additional value of plotting weighted defects? In what way does it provide an additional perspective than the defect classification trend?

Apply

1. Compute the cyclomatic complexity for the following program:

```
/*  SelectionSorter.java */
    public class SelectionSorter
    {
      public void sort(int [] A)
```

```
        {
            for (int i = A.length - 1; i > 0; i--)
            {
                int maxIndex = i;
                for (int j = i - 1; j >= 0; j--)
                {
                    if (A[maxIndex] < A[j])
                        maxIndex = j;
                }
                int temp = A[i];
                A[i] = A[maxIndex];
                A[maxIndex] = temp;
            } // for
        } // sort
    } // SelectionSorter
```

2. Calculate the cyclomatic complexity by writing a program to find whether a Triangle is isosceles/scalene /equilateral through identifying the Independent paths, Predicate nodes using Graph Matrices.
3. Compute MC/DC coverage for the above program.
4. For a Library Management system, draw a Cause-Effect graph explaining its way of test execution.
5. For the given program, identify the metric and calculate its value for computing its complexity using all types.

```
class Loopit {
  public static void main (String[] args) throws IOException {
    BufferedReader stdin = new BufferedReader
      (new InputStreamReader (System.in));
    int count, max, num;
    num = 0; // Assign initial value of count
    while (num != -1) {
      System.out.print ("Enter a number to factorialize (-1 to quit): ");
      System.out.flush();
      num = Integer.parseInt (stdin.readLine());
      max = 1; // Assign to 1, so factorial isn't zero every time
      if (num == -1) {
        System.out.println("Okay, quiting...");
```

```
}
else { // Since they're not quitting we better factorialize
  for  (count = 1; count<=num; count++) {
    max = count * max;
  }
  System.out.println (num+"! (factorial) is : "+ max);
}   }   } // method main   }
```

6. Compute the LCSAJ for the following code and generate test cases that are adequate to LCSAJ coverage.

```
1.  int x,y,p;
2.  Input(x,y)
3.  P=g(x)
4.  If(x<0)
5.  P = g(y)
6.  If (p=0)
7.  q=g(x)
8.  else
9.  q=g(x*y)
10. end
```

Create

1. Draw the CFG and generate test cases for the following program.

```
int compute_gcd(x, y)
int x, y;
{
while (x! = y){
if (x>y) then
x= x – y;
else y= y – x;
 }
 return x;
}
```

2. 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.
3. Using the equivalence partitioning method, construct a set of test cases for above program

4. Construct a control flow graph for the string a*abb*c+abdc*. Calculate the Cyclomatic complexity from the above graph using all types.
5. The department of public works for a large city decided to develop a Web-based Pothole tracking and repair system. The description is as follows:

Citizens can log onto a web site and report the location and severity of potholes. As potholes are reported they are logged within a "public works department repair system" and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb etc.) district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and includes pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and dollar amount of damage.

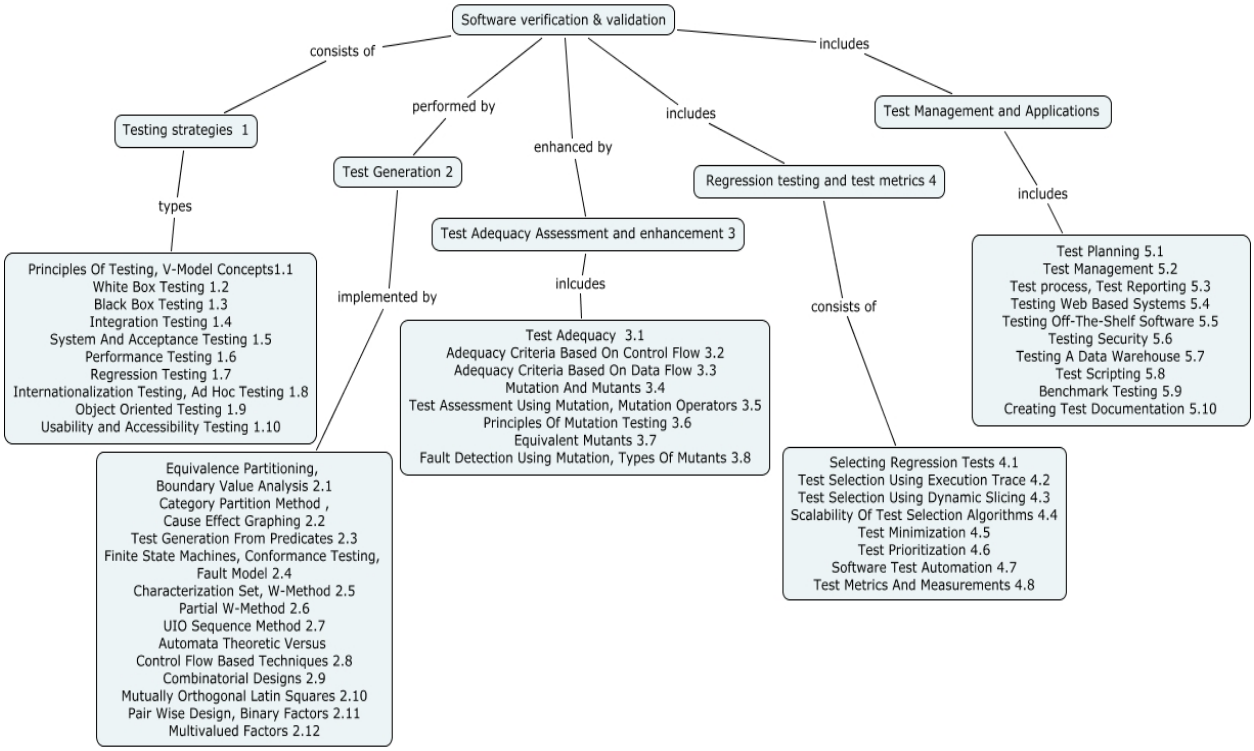
Using Boundary value analysis, construct a set of valid and invalid test cases for the above system.

6. Design a set of test cases using CFG and DFG for the following program that selects largest of three numbers.

```
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);
        }}
```

7. Consider a date validation program. Assuming that you have no access to the code, develop a set of test cases using any of the black box techniques. Present your result in the form of a table.

Concept Map



Syllabus

Testing strategies and techniques: Principles of Testing, V-model concepts, white box testing, black box testing, integration testing, system and acceptance testing, performance testing, regression testing, internationalization testing, ad hoc testing, object oriented testing, Usability and Accessibility Testing.

Test Generation: Equivalence partitioning, boundary value analysis, category partition method, cause effect graphing, test generation from predicates. Finite state machines, conformance testing, fault model, characterization set, W-method, partial W-method, UIO sequence method, automata theoretic versus control flow based techniques, combinatorial designs, mutually orthogonal Latin squares, pair wise design, binary factors, multivalued factors.

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, fault detection using mutation, types of mutants.

Regression testing and test metrics: Selecting regression tests, test selection using execution trace, test selection using dynamic slicing, scalability of test selection

algorithms, test minimization, test prioritization, Software test automation, test metrics and measurements.

Test Management and Applications: Test Planning, Test Management, Test process, Test Reporting, testing web based systems, testing off-the-shelf software, testing security, testing a data warehouse, Test Scripting, Benchmark Testing, creating test documentation.

Text Books :

1. Srinivasan Desikan, Gopalaswamy Ramesh, "Software testing – principles and practices", First Edition ,Pearson Education, 2009
2. Aditya P. Mathur "Foundations of Software Testing", First Edition ,Pearson Education, 2008

Reference Books:

1. William E. Perry, "Effective methods for software testing", Second Edition, John wiley & Sons,2000
2. Roger S. Pressman, Software Engineering A Practitioner's Approach, Seventh Edition, Mcgraw Hill International Edition.,2010

Course Content and Lecture Schedule

S.No.	Topic	No of Lectures
1.	**Testing strategies and techniques (10)**	
1.1	Principles Of Testing, V-Model Concepts	1
1.2	White Box Testing	1
1.3	Black Box Testing	1
1.4	Integration Testing	1
1.5	System And Acceptance Testing	1
1.6	Performance Testing	1
1.7	Regression Testing	1
1.8	Internationalization Testing, Ad Hoc Testing	1
1.9	Object Oriented Testing	1
1.10	Usability and Accessibility Testing	1
2.	**Test Generation (12)**	

2.1	Equivalence Partitioning, Boundary Value Analysis	1
2.2	Category Partition Method ,Cause Effect Graphing	1
2.3	Test Generation From Predicates	1
2.4	Finite State Machines, Conformance Testing, Fault Model	1
2.5	Characterization Set, W-Method	1
2.6	Partial W-Method	1
2.7	UIO Sequence Method	1
2.8	Automata Theoretic Versus Control Flow Based Techniques	1
2.9	Combinatorial Designs	1
2.10	Mutually Orthogonal Latin Squares	1
2.11	Pair Wise Design, Binary Factors	1
2.12	Multivalued Factors.	1
3.	**Test Adequacy Assessment and enhancement (8)**	
3.1	Test Adequacy Basics	1
3.2	Adequacy Criteria Based On Control Flow	1
3.3	Adequacy Criteria Based On Data Flow	1
3.4	Mutation And Mutants	1
3.5	Test Assessment Using Mutation, Mutation Operators	1
3.6	Principles Of Mutation Testing	1
3.7	Equivalent Mutants	1
3.8	Fault Detection Using Mutation, Types Of Mutants	1
4	**Regression testing and test metrics (10)**	
4.1	Selecting Regression Tests	1
4.2	Test Selection Using Execution Trace	1
4.3	Test Selection Using Dynamic Slicing	1

4.4	Scalability Of Test Selection Algorithms	1
4.5	Test Minimization	1
4.6	Test Prioritization	2
4.7	Software Test Automation	2
4.8	Test Metrics And Measurements	1
5	**Test Management and Applications (10)**	
5.1	Test Planning	1
5.2	Test Management	1
5.3	Test process, Test Reporting	1
5.4	Testing Web Based Systems	1
5.5	Testing Off-The-Shelf Software	1
5.6	Testing Security	1
5.7	Testing A Data Warehouse	1
5.8	Test Scripting	1
5.9	Benchmark Testing	1
5.10	Creating Test Documentation	1
	Total	**50**

Course Designers

1. A.Askarunisa aacse@tce.edu
2. A. Malini amcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C64	4	0	-	4

C64 Cryptography 4:0

Preamble: Security Implementations and Issues is a discipline which studies the theoretical, practical and managerial aspects of securing Information against threats. It utilises techniques from discrete applied mathematics, including algebra and probability, as well as from computer science, communications engineering and business. The course will enable the students to understand, develop, and deploy countermeasures to mitigate the risks inherent in the transmission, storage and retrieval of sensitive information.

Prerequisite: C42, C54

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

1. What are the threats and vulnerabilities of information systems including databases, networks, applications, internet-based communication, web services, and mobile technologies.
2. Determine the measures that protect and defend information and information systems by ensuring their authentication and authorization
3. Explain techniques to protect the data from attackers by providing confidentiality.
4. Select methods to ensure non-repudiation for the data to be used in corporates such as stock markets, banking etc.
5. Determine the strength of a given security algorithm when used in real time applications.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	10	10
2	Understand	20	20	10
3	Apply	40	50	60
4	Analyze	20	20	20
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember:

1. What is Crypt analysis?
2. Explain Brute force attack with an example.
3. Depict by a table the relationship between security and mechanisms.
4. Define Eulers totient function.
5. Define Discrete logarithm problem.
6. What is a finite field?

Understand:

1. Use sieve of Eratosthenes to generate prime numbers less than 200.
2. How does public key cryptosystem provide authentication?
3. What are the various types of crypt analytic attacks on encrypted systems?
4. What is the F function? Illustrate by means of a layout diagram the computations involving the 4 S-Box values in blow fish algorithm.
5. What is the strength of symmetric key over public key Cryptosystems.
6. Can the following matrix be used as key in Hill cipher. Justify your answer

   1 2 3
   4 5 6
   7 8 9

Apply

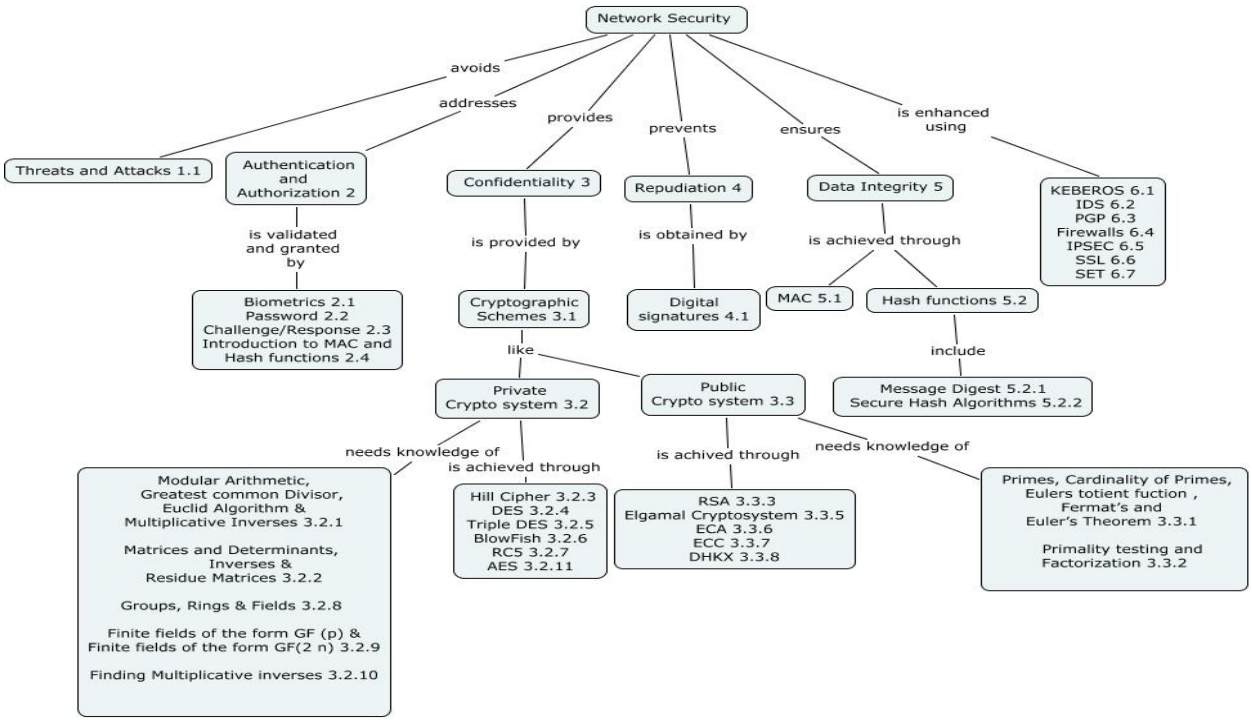
1. Apply Vignere cipher to message DONKEY with key =”DIVINE”
2. Perform encryption and decryption using RSA algorithm for the following:
   a. n=33, M=5 (b) n=55, M=9 (c) n=77, M=8 (d) n=143, M=7 (e) n=527,M=2
3. Apply the key expansion pseudo code and generate the first 4 words of the key using the key byte values in AES.
4. Given E67(2,3), e1=(2,22), d=4, determine e2=d.e1, by first multiplying and then adding twice.. Applying El Gamel principle evaluate C1=r.e1 and C2=P+r.e2, Where P is the message (24,26) At the receiving end using C1 and C2, recover the plain text.
5. For E13(1,1), What is 4P, given P=(1,4)?
6. What is the number of mappings (Keys) in mono alphabetic cipher if the language used is Tamil?

Analyze

1. Use the relative frequency of letters in English to decipher the following cipher text: UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZVUEP HZHMDZSHZOWSSFPAAPPDTSVPQUZWYMXUZUHSXSPYEPOPDZSZU FPOMBZWPFUPZHMDJUDTMOHMQ

2. Evolve an expression for the expected running time of the attack given key is 56-bit, message is 64-bit and the number of trials is n.
3. Analyze and identify the potential location of confidentiality attack on a LAN.
4. Get the plaintext from the cipher "HQJLQHHULQJ" obtained Caeser cipher by frequency analysis. (Hint: Most frequently occurring letters are E and S in English).
5. Evolve the 8x8 transformation matrix using the mapping function. bi'=b(i+4)mod8+b(i+5)mod8+b(i+6)mod8+b(i+7)mod8+ci for forward transformation of state matrix elements.
6. Hill cipher succumbs to a known plaintext attack if sufficient plaintext-ciphertext pairs are provided. It is even easier to solve the Hill cipher if a chosen plaintext attack can be mounted. Analyze and identify such attacks.
7. In one of Dorothy Sayer's mysteries, Lord Peter is confronted with the message SDEEEFRGGRFVBHTBNHTYJUNJGSD, he also discovers the key to the message, which is a sequence of integers: 78495873854. Try to identify what may be the message. How secure is the scheme if the algorithm is known and not the key?

Concept Map



Syllabus:

Introduction- Attacks and Threats on Networks, **Security Services: Authentication and Authorization** - Biometrics, Passwords, Challenge Response protocols, Introduction to MAC and Hash Functions, **Confidentiality –**Cryptography-

Mathematics behind Cryptography – Number Theory, Modular Arithmetic, Greatest Common Divisor, Multiplicative Inverses, Groups, Rings, Fields, Finite Fields of the form GF (p), Finite fields of the form GF(2^n), Primes, Cardinality of Primes, Eulers totient fuction , Fermat's and Euler's Theorem, Primality Testing and Factorization, primitive roots, Discrete logarithm **- Symmetric Key Crypto systems** – Hill Cipher, Data Encryption standard, Triple DES, Blowfish, RC5 Advanced Encryption Standard. **Public key cryptosystems** - RSA, Diffie Hellman Key Exchange, ElGamal Cryptosystem, Elliptic Curve Cryyptography, **NonRepudiation** – Digital Signatures, **Data Integrity -** Message Authentication code, Hash functions – Message Digest, Secure Hash Algorithm, **Applications –** Kerberos,PGP, IP Sec, IDS, Firewalls, Secure Socket Layer, Secure Electronic Transaction.

Textbook

1. W. Stallings ," Cryptography and Network Security: Principles and Practice", PHI 4th Edition
2. Behrouz A.Foruzan, "Cryptography and Network Security", Tata McGraw Hill 2007
3. Nina Gobbole: Information Systems Security: Security management, Metrics, Framework and best practices, 1st Edition, Wiley, 2009

Reference Book

1. Mark Stamp: Information Security Principles and Practice, Wiley-Inter science, 2001

Course Contents and Lectures schedule

No	Topic	No of Lectures
1	**Introduction (1)**	
1.1	Attacks and Threats	1
2	**Authentication and Authorization (5)**	
2.1	Biometrics	2
2.2	Passwords	1
2.3	Challenge Response Schemes	1
2.4	Introduction to MAC and Hash Functions	1
3	**Confidentiality (31)**	
3.1	Cryptographic Schemes - Introduction	1

3.2	Private Crypto system	1
3.2.1	Modular Arithmetic, Greatest common Divisor, Euclid Algorithm, Multiplicative Inverses	2
3.2.2	Matrices and Determinants, Inverses, Residue Matrices	2
3.2.3	Hill Cipher	1
3.2.4	Data Encryption Standard	2
3.2.5	Triple DES	1
3.2.6	Blow Fish	2
3.2.7	RC5	1
3.2.8	Groups, Rings, Fields	1
3.2.9	Finite fields of the form GF (p), Finite fields of the form GF(2 n),	2
3.2.10	Finding Multiplicative inverses,	1
3.2.11	Advanced Encryption Standard	2
3.3	Public Key Cryptosystems	1
3.3.1	Primes, Cardinality of Primes, Eulers totient fuction , Fermat's and Euler's Theorem	2
3.3.2	Primality testing and Factorization	2
3.3.3	RSA	1
3.3.4	Primitive roots, Discrete Logarithm, Diffie Hellman Key Exchange	1
3.3.5	Elgamal Cryptosystem	1
3.3.6	Elliptic Curve Arithmetic	2
3.3.7	Elliptic Curve Cryptography	1
3.3.8	Diffie Helman Key Exchange	1
4	**Non-Repudiation (1)**	
4.1	Digital Signatures	1
5	**Message Authentication and Integrity (6)**	
5.1	MAC	1

5.2	Hash Functions	2
5.2.1	Message Digest	2
5.2.2	Secure Hash Algorithms	1
6	**Applications (7)**	
6.1	Kerberos	1
6.2	Intrusion Detection System	1
6.3	Pretty Good Privacy	1
6.4	Firewalls	1
6.5	IP Security	1
6.6	Secure Socket Layer	1
6.7	Secure Email Transfer	1
	Total	**50**

Course Designer:

1. Dr. P. Subathra pscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C67	-	-	3	1

C67 Cryptography Laboratory 0:1

Preamble: To provide the knowledge in the various security algorithms and thus enable the students to familiarize with security programming.

Prerequisite: C47

List of Exercises:

CYCLE I

1. Implement Substitution and Transposition Techniques
2. Implement Hill Cipher using 3 x 3 key matrix
3. Implement simplified DES Algorithm
4. Implement DES Algorithm
5. Implement Kerberos Algorithm
6. Implement AES Algorithm

CYCLE II

7. Implement RC4 Stream cipher algorithm
8. Implement Diffie Hellman key exchange Algorithm
9. Implement Elliptic Curve Arithmetic Encryption and Decryption algorithm
10. Implement Message Authentication codes
11. Implement Secure Hash Algorithm
12. Implement pseudo number generator algorithm

Course Designers

1.M.P.Ramkumar mpr@tce.edu

2.Dr.P.Subathra pscse@tce.edu

3. C.Sridharan cscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C68	-	-	3	1

C68 Software Testing Laboratory 0:1

Preamble: This laboratory course encourages students to practice the disciplined approach of software testing by giving an exposure to various software testing tools that enhances the quality of software

Prerequisite: C63

List of Exercises:

1. Test case generation and execution using JUnit tool.
2. Test Case Coverage measurement using Codecover, EclEmma, Clover tools.
3. Mutation testing using Muclipse tool.
4. GUI testing using Abbot tool
5. Perform the following tests using Win runner tool
    - i) Synchronizing Tests
    - ii) Checking GUI objects
    - iii) Checking Bitmaps
    - iv) Programming Tests with TSL
    - v) Reading and verifying Text
    - vi) Batch Tests

A possible set of applications may be the following:

a. E-Library System
b. Inventory System
c. Course Registration System
d. Quiz System
e. Student Marks Analyzing System
f. Reservation Systems for Airlines, Railways etc.
g. Real-Time Scheduler

Course Designers

1. A.Askarunisa aacse@tce.edu
2. A.Malini amcse@tce.edu

42nd meeting of the Academic Council Resolution No.42 :10

Annexure-H

REVISED CURRICULUM AND DETAILED SYLLABI

FOR

B.E DEGREE (Computer Science and Engineering) PROGRAM

SEVENTH AND EIGHTH SEMESTERS

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2013-2014 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided ISO 9001-2000 certified
Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

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

Department of Computer Science and Engineering

Graduating Students of BE program of Computer Science and Engineering will be able to

1. Specify, design, develop, test and maintain usable software systems that behave reliably and efficiently and satisfy all the requirements that customers have defined for them
2. Work in a team using common tools and environments to achieve project objectives
3. Develop software systems that would perform tasks related to Research, Education and Training and/or E-governance

Thiagarajar College of Engineering, Madurai-625015

Department of Computer science and Engineering

Scheduling of Courses for those who are joined in the academic year 2013-2014

Semester	Theory						Laboratory/Project		
8th (21)	Elective 6 3:0	Elective 7 3:0	Elective 8 3:0					C88 Project 0:12	
7th (21)	C71 Software Project and Quality Management 3:0	C72 Accounting and Finance 3:0	Elective 3 3:0	Elective 4 3:0	Elective 5 3:0			C78 Project 0:6	
6th (22)	C61 Management Theory & Practice 3:0	C62 Theory of Formal Languages 3:0	C63 Software Verification & Validation 3:1	C64 Cryptography 4:0	Elective 1 3:0	Elective 2 3:0	C67 cryptography Lab 0:1	C68 Software Testing Lab 0:1	
5th (24)	C51 Statistics And Graph Theory 4:0	C52 Databases Principles And Design 3:1	C53 Multicore architectures 3:1	C54 Computer Networks: Principles 3:0	C55 Web Programming 3:0	C56 Software Design 3:1	C57 Web Programming Lab 0:1	C58 Software Design Lab 0:1	
4th (26)	C41 Discrete Mathematics 4:0	C42 Computer Networks: Use & Configuration 3:0	C43 Principles of Compiler Design 3:0	C44 Operating Systems: Principles and Design 3:1	C45 Software Engineering 3:1	C46 Data Bases: Practice 3:1	C47 Network Programming Lab 0:1	C48 Data Bases Lab 0:1	C49 Professional Communications 1:1
3rd (23)	C31 Engineering Mathematics– 3 4:0	C32 Graphics and visualization 3:0	C33 Systems Software 4:0	C34 Operating Systems: Configuration & Use 3:0	C35 Design and Analysis of Algorithms 3:0	C36 Object Oriented Programming 3:1	C37 Operating Systems and System Software Lab 0:1	C38 Object Oriented Programming Lab 0:1	
2nd (22)	C21 Engineering Mathematics-2 4:0	C22 Digital Systems 3:1	C23 Computer Organization 3:0	C24 Procedural Programming 3:0	C25 Data Structures 3:0	C26 Environment and Ecology 2:0	C27 Data Structures Lab 0:1	C28 Procedural Programming Lab 0:1	C29 Workshop 0:1
1st (25)	H11 Engineering Mathematics – 1 4:0	H12 Physics 3:0	H13 Chemistry 3:0	H14 English 3:0	H15 Basics of ME and CE 4:0	H16 Basics of EEE 4:0	H17 Physics Lab 0:1	H18 Chemistry Lab 0:1	H19 Engineering Graphics 0:2

Board of Studies Meeting on 30.04.2011 Approved in 42nd Academic Council Meeting on 11.06.11

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

B.E Degree (Computer Science and Engineering) Program

SUBJECTS OF STUDY

(For the candidates admitted from 2013-2014 onwards)

SEVENTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
C 71	Software Project and Quality Management	DC	3	-	-	3
C 72	Accounting and Finance	DC	3	-	-	3
CCX	ELECTIVE 3	DE	3	-	-	3
CCX	ELECTIVE 4	DE	3	-	-	3
XGX	ELECTIVE 5	GE	3	-	-	3
PRACTICAL						
C78	Project	DC	-	-	12	6
Total			**15**	**-**	**12**	**21**

EIGHTH SEMESTER

Subject code	Name of the subject	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
CCX	Elective 6	DC	3	-	-	3
CCX	Elective 7	DC	3	-	-	3
CCX	Elective 8	DC	3	-	-	3
PRACTICAL						
C88	Project	DC	0	-	24	12
Total			**9**	**0**	**24**	**21**

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

Note:
1 Hour Lecture/Tutorial is equivalent to 1 credit
2/3 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 625 015
B.E Degree (Computer Science and Engineering) Program
SCHEME OF EXAMINATIONS
(For the candidates admitted from 2013 - 2014 onwards)

SEVENTH SEMESTER

S.No.	Sub. Code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	C 71	Software Project and Quality Management	3	50	50	100	25	50
2	C 72	Accounting and Finance	3	50	50	100	25	50
4	CCX	ELECTIVE III	3	50	50	100	25	50
5	CCX	ELECTIVE IV	3	50	50	100	25	50
6	XGX	ELECTIVE V	3	50	50	100	25	50
PRACTICAL								
6	C78	Project	-	150	150	300	75	150

EIGHTH SEMESTER

S. No	Sub. Code	Name of the subject	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Termin al Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	CCX	Elective 6	3	50	50	100	25	50
2	CCX	Elective 7	3	50	50	100	25	50
3	CCX	Elective 8	3	50	50	100	25	50
PRACTICAL								
4	C88	Project	-	150	150	300	75	150

* Continuous Assessment evaluation pattern will differ from subject to subject and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

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

Sub Code	Lectures	Tutorial	Practical	Credit
C71	3	-	-	3

C71 Software Project and Quality Management 3:0

Preamble: Software Project and Quality Management comprises of processes that ensure that the Software Project would reach its goals. In other words software projects which are planned, monitored and controlled would meet the client's expectation. Improving software quality can give a quality product which meets its requirements and satisfies the user

Prerequisite: C45 Software Engineering, C56 Software Design, C63 Software Verification & Validation

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve Engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

At the end of the course, the student will be able to

1. Understand Quality in Business scenario.
2. Understand Software process Management methodologies.
3. Analyze and identify Quality Standards and Guidelines
4. Ability to manage Metrics.
5. Identify techniques for Project Control
6. Determine Software Quality Management Models
7. Apply Metrics for continuous improvement of quality

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	10	10	20
2	Understand	40	30	50
3	Apply	0	10	30
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. What is software management?
2. What is software economics?
3. What is software cost estimation?
4. What are the three levels of process?
5. What are pear inspections?
6. What are the two stages of life cycle?

Understand

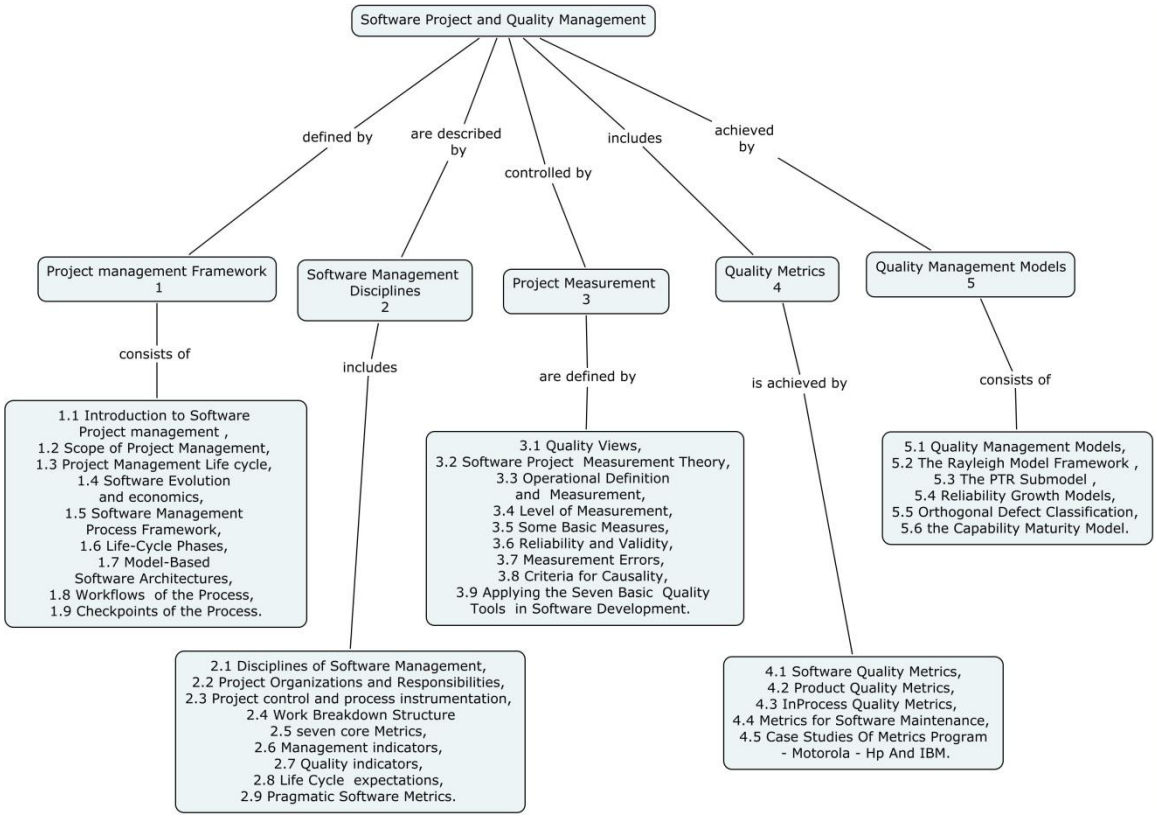
1. How to reduce software size?
2. How to improve software Process?
3. How to improve team effectiveness?
4. Explain the life cycle phases of the project.
5. Compare the technical and management perspective of architecture.
6. Explain the software process workflows.

Apply

1. How Rayleigh mode can be used for projecting the latent software defects when the development work is complete and the product is ready to ship to customers
2. Apply cause and effect diagram and paretto diagrams can be used to identify dominant problems and their root causes in software development process
3. Derive the PTR model curve for the following defect discovery pattern which was observed for the first release of an operating system: Month 1: 17%, Month 2: 22%, Month 3: 20%, Month 4: 16%, Month 5: 12%, Month 6: 9%, Month 7: 4%

4. Develop a defect tracking and reporting system and a set of related in-process metrics to implement Quality management models.
5. How to implement The Rayleigh model Recommendations for Small Organizations that don't have data and metrics tracking for all phases of development?

Concept Map



Syllabus

Project management Framework: Introduction to Software Project management and control, Scope of Project Management, Project Management Life cycle, Software Evolution and economics, Software Management Process Framework, Life-Cycle Phases, Model-Based Software Architectures, Workflows of the Process, Checkpoints of the Process; **Software Management Disciplines:** Disciplines of Software Management, Project Organizations and Responsibilities, Project control and process instrumentation, Work Breakdown Structure, seven core Metrics, Management indicators, Quality indicators, Life Cycle expectations, Pragmatic Software Metrics; **Project Measurement:** Quality Views, Software Project Measurement Theory, Definition, Operational Definition and Measurement, Level of

Measurement, Some Basic Measures, Reliability and Validity, Measurement Errors, Criteria for Causality, Applying the Seven Basic Quality Tools in Software Development; **Quality Metrics:** Software Quality Metrics, Product Quality Metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Case Studies Of Metrics Program - Motorola - Hp And IBM; **Quality Management Models:** The Rayleigh Model, Quality Management Models, The Rayleigh Model Framework , The PTR Submodel , Reliability Growth Models, Orthogonal Defect Classification, the Capability Maturity Model, Levels of CMM, CMMI.

Text Books :

1. Walker Royce, "Software Project Management A Unified Framework",Pearson Education, 2004.
2. Stephen H. Kan, Metrics and Models in Software Quality Engineering, Second Edition, Addison Wesley,2002

Reference Books:

1. Bob Hughes, Mike Cotterell, "Software Project Management" Tata Mcgraw-Hill 2010.
2. Norman E – Fentar, Share Lawrence Pflieger, "Software Metrics", International Thomson Computer Press, 1997.
3. Alan C Gillies, "Software quality: Theory and Management", second edition, Thomson Learning, 2003.

Course Contents and Lecture Schedule

S.No.	Topic	No of Lectures
1.	**Project management Framework**	
1.1	Introduction to Software Project management and control	2
1.2	Scope of Project Management	2
1.3	Project Management Life cycle	2
1.4	Software Management Process Framework	2
1.5	Workflows of the Process. Checkpoints of the Process.	2
2.	**Software Management Disciplines**	
2.1	Project Organizations and Responsibilities	1
2.2	Project control and process instrumentation	2
2.3	Work Breakdown Structure	1

S.No.	Topic	No of Lectures
2.4	seven core Metrics	2
2.5	management indicators Quality indicators	2
2.6	Life Cycle expectations,	1
2.7	Pragmatic Software Metrics, Metrics automation	1
3.	**Project Measurement**	
3.1	Quality Views,	2
3.2	Measurement Theory	1
3.3	Level of Measurement , Some Basic Measures,	2
3.4	Reliability and Validity, Measurement Errors,	2
3.5	Criteria for Causality,	1
3.6	Applying the Seven Basic Quality Tools in Software Development.	2
4	**Quality Metrics**	
4.1	Software Quality Metrics	2
4.2	Product Quality Metrics ,	2
4.3	In-Process Quality Metrics,	2
4.4	Metrics for Software Maintenance,	2
4.5	Case Studies Of Metrics Program - Motorola - Hp And IBM.	2
5	**Quality Management Models**	
5.1	The Rayleigh Model	2
5.2	Quality Management Models,	1
5.3	The Rayleigh Model Framework	1
5.4	The PTR Submodel,	2
5.5	Reliability Growth Models,	1
5.6	Orthogonal Defect Classification	1
5.7	the Capability Maturity Model, Levels of CMM, CMMI.	2
	Total	**50**

Course Designer

1. N.Shivakumar shiva@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C72	3	-	-	3

C72 Accounting and finance 3.0

(Common to Mechanical : G56)

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 things data about the organizations 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.

Program outcomes addressed

- Engineering graduates will understand the basic concepts, processes, tools and techniques of accounting and finance.
- Engineering graduates will apply the concepts, processes, tools and techniques of accounting and finance and take effective decisions in organizational settings.

Competencies:

At the end of the course, the students will be able to

1. Develop an understanding about what accounting is and its importance in decision making.
2. Understand the recording function of accounting.
3. Understand the classification function of accounting.
4. Understand the summarizing function of accounting.
5. Understand the analysis and interpretation function of accounting.
6. Perform the various functions of accounting.
7. Prepare trial balance and there from financial statements like trading account, Profit & loss account and balance sheet.
8. Interpret the financial statements of an organization.
9. Understand the meaning of financing and its functions and objectives.
10. Understand some of the basic concepts, tools and techniques of finance and their applications.

Assessment pattern

	Bloom's category	Test 1	Test 2	Test 3/End-Semester examination
1	Remember	20	20	20
2	Understand	30	30	30
3	Apply	50	50	50
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. The meaning of the term "accounting" and definition of accounting
2. The concepts and conventions of accounting.
3. Its importance in decision making.
4. The functions of accounting
5. The meaning of the term Depreciation and causes of Depreciation
6. The meaning of the term "Cost accounting"

Understanding

1. Understand the definition of accounting
2. Explain the various functions of accounting.
3. Discuss the concepts and conventions of accounting.
4. Understand the process of preparing final accounts.
5. Understand the concept of depreciation and methods of providing depreciation
6. Explain the classification of cost.

Apply

1. Journalise the following business transactions:
   A) A brings in cash Rs.10, 000 as the capital and purchases land worth Rs.2000.
   b) He purchases goods worth Rs.5, 000.
   C) He returns goods worth Rs.500 as they are defective.
   d) He sells goods for Rs.7, 000.
   e) He incurs traveling expenses of Rs.200.
2. Record and classify the following transactions in the books of Suresh
   Suresh introduces capital of Rs.20, 000 into his business.
   He purchases furniture worth Rs.2000.
   He purchases goods worth Rs, 8,000.

He incurs Rs.200 as freight expenses.

He sold goods for cash Rs.5, 000 and for credit Rs.2000

He paid salary Rs.3, 000

He paid electricity expenses Rs.800.

3. Prepare Trading and profit and loss account and Balance sheet on 31.12.96 from the following trial balance extracted from the books of Mr. Kumar as on 31.12.96.

Debit Balances	Rs.	Credit Balances	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 and doubtful debts	600
Sundry debtors	30,000		
General expenses	1,600		
Cash at bank	9,400		
Rates and taxes	1,200		
Bad debts	400		
Insurance premium	800		
Discount allowed	1,400		
Opening stock	20,000		
Total	3,33,200		3,33,200

4. Senthil purchased machinery for Rs.4, 00,000 on 1st April 2000.On 1st April 2001 additional machinery was purchased for Rs.40, 000.prepare the asset account for three years. Depreciation is to be provided at 10%p.a using straight line method. The firm closes its books on 31st March of every year.

5. A factory is currently working at 50% capacity and the product cost is Rs.180 per unit as below:

Material ---------- Rs.100

Labor ---------- Rs.30

Factory overhead— 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%.

6**.** The following particulars are extracted from the books of a company relating to commodity "A" for the half year ending 30th June1993.

Purchase of raw materials ---------- Rs.1, 32,000
Direct wages ---------- Rs.1, 10,000
Rent, rates, insurance and works cost-------- Rs.44, 000
Carriage inward --------- Rs.1584
Stock on 1-1-93
Raw materials---------- Rs.22, 000
Finished product (1600 tones) - Rs.17, 000
Stock on 30-6-93
Raw materials ---------- Rs.24, 464
Finished products (3,200 tones) ---------- Rs.35, 200
Work-in-progress on 1-1-93 ---------- Rs.17, 600
Work-in-progress on 30-6-93 ------ Rs.5280
Factory supervision ---------- Rs.8, 800
Sales-Finished products ---------- Rs.33, 000

Advertising discount allowed and selling cost at Re.0.75per tones sold.25, 600 tones of commodity was sold during the period.

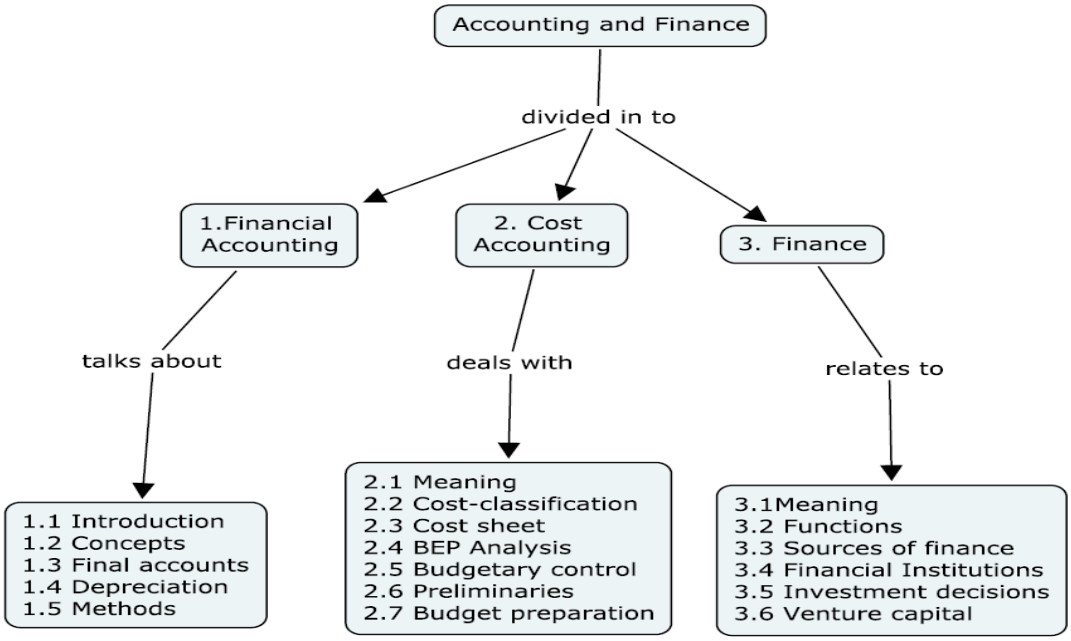
You are required to ascertain:

i) prime cost ii) factory cost iii) cost of sales iv) profit v) No of tones of the commodity sold.

7. From the following information calculate the Break even point in terms of units and Break even point in terms of sales

Sales in Rs.10, 000, Variable costs (direct material and direct labor) Rs.6, 000, Fixed cost Rs.2, 000 profit Rs.2, 000, No. of units produced 1,000 units.

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 for providing depreciation; **Cost Accounting:** Meaning and importance-Cost-Elements of Cost-cost Classification-Preparation of cost sheet-Material costing-Valuation of purchases-pricing of material issues. Break-even analysis-managerial applications. Budgetary control-Introduction-objectives of budgetary control-preliminaries for operation of budgetary control-Budgets-types of budgets and their preparation; **Finance:** Meaning-Definition-Objectives-functions of finance-source of finance-short-term, Long-term and medium-term-Role of special financial institution in financing-Investment decisions-short-term Investments and long-term investments-Venture Capital.

Text Books

1. M.c.Shukla, T.s.Grewal, S.c.Gupta: "Advanced Accouts-volumeI", 2007 Reprint, S.Chand&Company Ltd.2007.

2. S.P.Jain, K.L.Narang: "Advanced Accountancy-Volume I", Thirteenth Revised Edition, Kalyani Publishers.2006.

3. V.K.Saxena, C.D.Vashist: "Advanced cost and Management Accounting", Seventh Enlarged Edition, Sultan Chand and Sons, 2008.

Reference Books

1. Prasanna Chandra, "Financial Management-Theory and Practice" Sixth Reprint, Tata McGraw-Hill publishing company Limited, 2007.
2. Ramachandra Aryasri, A, Ramana Moorthy, V.V, Engineering Economics and financial Accounting", Tata McGraw hill, 2007.
3. S.N.Maheswari, "Advanced accountancy"Vikas publishing, 2007.

Course content and lecture schedule

S.No	Topics	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 &Loss account and Balance sheet.	6
1.4	Depreciation –Meaning-Need and objectives	2
1.5	Basic factors-Methods for providing depreciation	3
2.	**Cost Accounting**	
2.1	Meaning and importance	2
2.2	Cost-Elements of cost-Cost classification	2
2.3	Preparation of Cost sheet-Material costing-valuation of purchases-pricing of material issues.	6
2.4	Break-even analysis-managerial applications	2
2.5	Budgetary control-introduction-objectives of budgetary control	1
2.6	Preliminaries for operation of budgetary control	1
2.7	Budget-Types of budgets and their preparation	4
3	**Finance**	
3.1	Meaning-Definition-objectives	2
3.2	Functions of finance	1
3.3	Source of finance-short-term, medium-term, long-term	2
3.4	Role of special financial institutions in financing	2
3.5	Investment decisions-Short-term investments and long-term investments	5
3.6	Venture capital	2

S.No	Topics	No. of Lectures
	Total	**46**

Course Designers:-

1. S. Dhanasekaran sdmech@tce.edu
2. P. S. Boopathi Manickam psbmeco@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCA	3	0	--	3

CCA Numerical Methods And Number Theory 3:0

Preamble: An engineering student needs to have some basic mathematical tools and techniques. 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 the numerical solutions in the field of polynomial and transcendental equations, simultaneous equations, interpolation, differentiation and integration, ordinary and partial differential equations and number theory.

Program Outcomes addressed

a. Graduate will demonstrate an ability to apply knowledge of Engineering and Information Technology in mathematics and Science.
b. Graduate will demonstrate an ability to identify, formulate and solve engineering problems.
c. Graduate will develop confidence for self education and ability to engage in life-long learning.

Competencies

At the end of the course the student should be able to:

1. Find greatest common divisor of two positive integers through algorithms for performing computer arithmetic using binary expansions.
2. Apply the concept of modular arithmetic in generating pseudorandom numbers, assigning computer memory location to files, encrypting and decrypting messages.
3. Apply the concept of counting in determining the complexity of algorithms.
4. Apply the concept of solutions of algebraic and transcendental equations in engineering problems by formulating such equations.
5. Apply the different techniques for getting the solution of a system of simultaneous equations using direct and iterative methods.
6. Identify the importance of Eigen values for a matrix and produce various techniques to find them.
7. Interpolate and extrapolate the given data using different methods of interpolation with the help of various operators.

8. Discuss the process of Numerical Integration to related problems of engineering and technology for getting approximate values of the given integral .
9. Formulate and give Numerical solutions using various techniques for ODEs modeled in engineering and technology.
10. Formulate and give Numerical solutions using various techniques for PDEs modeled in engineering and technology.

Assessment Pattern

	Bloom's category	**Test 1**	**Test 2**	**Test 3 / End Semester Examination**
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level learning objectives

Understand

1. Interpret the importance and significance of the Chinese Remainder theorem in Number theory.
2. Distinguish between r-combinations and r-permutations of a set with n elements.
3. Compare the exact solution and approximate solution of equations
4. Discuss the various techniques for the approximate solution of Algebraic and transcendental equations.
5. List the various methods for obtaining the approximate solution of system of simultaneous equations stating the basic principles used.
6. List the various method to solve ordinary differential equation and partial differential equation by numerical methods.

Apply

1. Show that a positive integer is divisibly by 3 if and only if the difference of the sum of its binary digits in even- numbered positions and the sum of its binary digits in odd-numbered positions is divisibly by 3
2. Demonstrate the pigeon-hole principle to show that among any 11 integers,

atleast two must have the same last digit.

3. Using Newton's method find the root of $x^3 - 4x^2 + x + 6 = 0$; $x_0 = 5$ correct to 4 decimal places

4. Solve the following system of equations by Gauss Jacobi method

$8x + y + z = 8$; $2x + 4y + z = 4$; $x + 3y + 3z = 5.$

5. Find the value of $y(0.2)$ *and* $y(0.4)$ using Runge-Kutta method of fourth order with h=0.2 given that $\frac{dy}{dx} = \frac{y^2 - x^2}{y^2 + x^2}$; $y(0) = 1.$

6. Solve : $u_t = u_{xx}$ given $u(0,t) = 0$; $u(x,0) = x(1-x)$; $u(1,t) = 0$; assume h=0.1 and choose suitable k so that u(i,j) is found out for i=0,0.1…1 and j=k,2k,3k.

Course contents and lecture schedule

No	Topic	No. of Lectures
1	**The fundamentals: Algorithms and the integers**	
1.1	The integers and division	3
1.2	Primes and Greatest Common Divisors	2
1.3	Integers and algorithms	2
1.4	Applications of Number Theory	3
2	**Counting**	
2.1	The Basics of Counting	2
2.2	The pigeon-hole principle	2
2.3	Permutaitons and Combinations.	2
2.4	Binomial Coefficients.	2
2.5	Generalized permutations and combinations	2
3	**Solutions to Algebraic and Transcendental equations**	
3.1	Bisection , Regula-falsi Method	2
3.2	Newton- Raphson method ,Iterative method	2
3.3	Horner's method	2
3.4	Graffe's root squaring method	2
3.5	Barstow's method	2

4	**Solution of simultaneous linear algebraic equations**	
4.1	Gauss elimination and Gauss Jordan methods	2
4.2	Crout's method, Gauss Jacobi and Gauss Siedal methods	2
4.3	Inversion by Gauss Jordan and Crout's methods	2
4.4	Relaxation Method	2
4.5	Power method for finding Eigen values	2
5	**Numerical Solution of ODE and PDE**	
5.1	Runge Kutta Method of fourth order	1
5.2	Predictor-Corrector Method- Adams Bashforth, Milne's Method	2
5.3	Boundary value problem – Solution by finite difference method	2
5.4	Classification of PDE – Finite difference approximations	1
5.5	Solution of Elliptic equations by Liebmann's Method	2
5.6	Solution of parabolic and Hyperbolic equations by explicit methods	2
	TOTAL	50

Syllabus:

The fundamentals: Algorithms and the integers: The integers and division, Primes and Greatest Common Divisors, Integers and algorithms, Applications of Number Theory. **Counting** : The Basics of Counting, The pigeon-hole principle , Permutations and Combinations, Binomial Coefficients, Generalized permutations and combinations. **Solutions to Algebraic and Transcendental Equations**: Bisection, Regula falsi, Newton-Raphson, Iterative Methods, Horner's Method, Giraeffes Root Squaring Method, Bairstow's method. **Solution of Simultaneous linear Algebraic Equations**: Gauss Elimination, Gauss Jordan, Crouts, Gauss Seidel, Gauss Jacobi, Inversion by Gauss Jordan and Crout's Method, Relaxation method, Power method for finding eigen values. **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

3. B.S.Grewal," Numerical Methods", Tata McGraw-Hill Publishing Company

Reference Books:

1. Robert.J Schilling, Sandra L.Harris "Applied Numerical Methods for Engineers Using Matlab and C" Thomson Books/cole,1999

2. Sastry S.S "Introductory Methods of Numerical Analysis" Prentice Hall of India -2006

Course Designers:

1. V.Mohan vmohan@tce.edu
2. M.Kameswari mkmat@tce.edu
3. N.Chitra ncmat@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCB	3	-	-	3

CCB Information Retrieval 3:0

Preamble: Information retrieval is concerned with techniques, strategies, representations, and search basics expected to fetch specific facts, answer questions, or compose reports that enable students to retrieve today's demanding information needs.

Prerequisite: C25, C35 .

Programme Outcomes addressed

b. Graduates will identify, formulate and solve engineering problems.

e. Graduates will able to work on multidisciplinary teams.

f. Graduates will use techniques, skills, modern engineering tools and software's to analyze problems

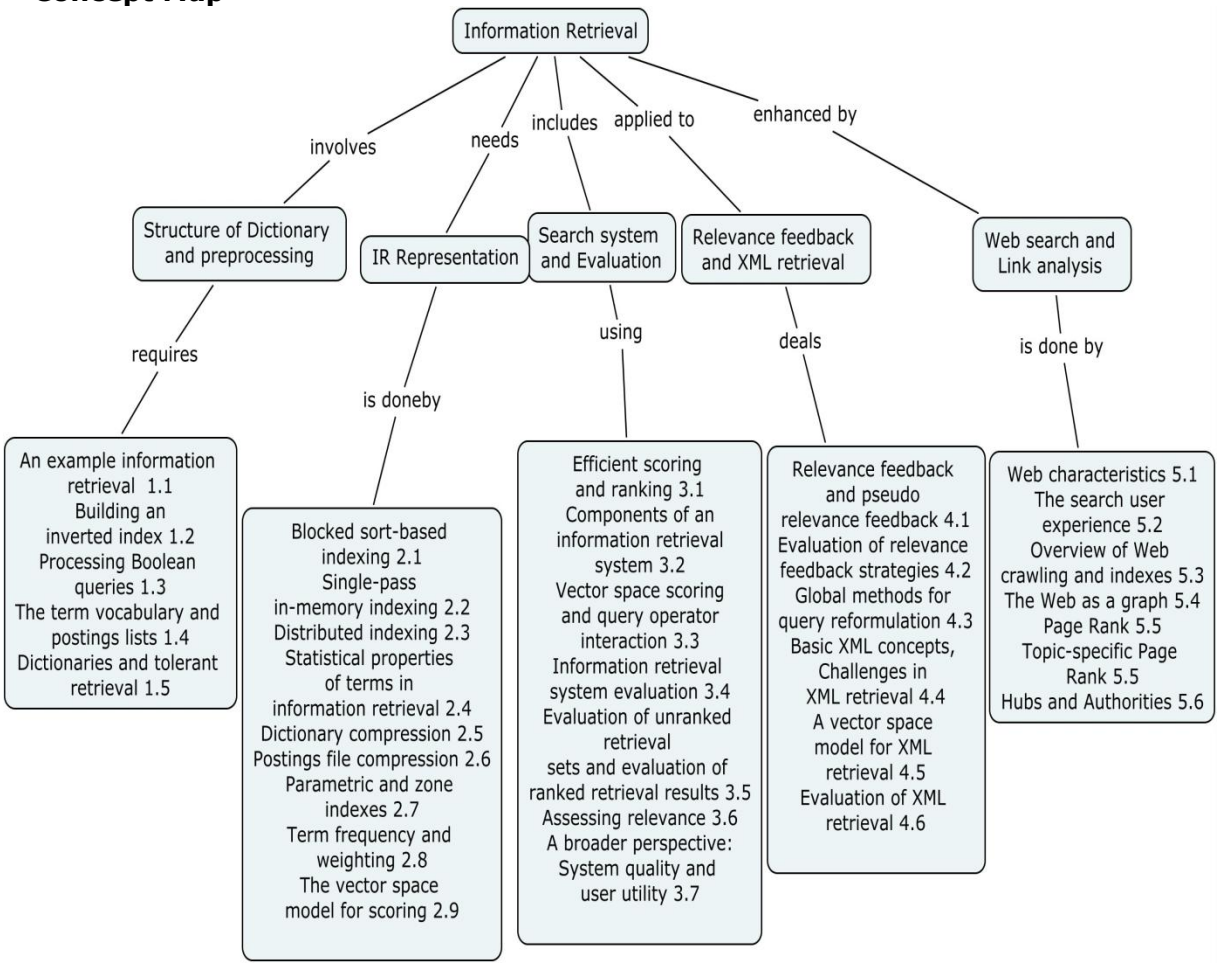
Competencies

1. Explain the principles of Information Retrieval
2. Understand the process of Information Retrieval design
3. Understand the process of search system and evaluation in information retrieval.
4. Apply the concept to XML retrieval.
5. Draw the inverted index for the given document collections.
6. Analyze the performance of IR system for the combine use of a positional index and stop words.
7. Apply Map Reduce to the problem of counting how often each term occurs in a set of given files.
8. Compute variable byte codes for the given numbers.
9. Analyze the reasons of relevance feedback has been little used in web search.
10. Analyze why positive feedback likely to be more useful than negative feedback to an IR system.
11. Illustrate the various components of web Search engine.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	20	10	10
3	Apply	40	40	40
4	Analyze	20	30	40
5	Evaluate	0	0	0
6	Create	0	0	0

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, 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, features a crawler must provide, features a crawler should provide, crawler architecture, the Web as a graph, anchor text and the web graph,PageRank,the Page Rank computation, topic-specific PageRank,hubs and authorities

Text Books :

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 Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Structure of Dictionary and preprocessing (10)**	
1.1	An example information retrieval problem	1
1.2	Building an inverted index	1
1.3	Processing Boolean queries	2
1.4	The term vocabulary and postings lists	3
1.5	Dictionaries and tolerant retrieval	3
2.	**IR Representation (10)**	
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	Parametric and zone indexes	1
2.8	Term frequency and weighting	1
2.9	The vector space model for scoring	2
3.	**Search system and Evaluation (10)**	
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	2
3.5	Evaluation of unranked retrieval sets and evaluation of ranked retrieval results	2
3.6	Assessing relevance	2
3.7	A broader perspective: System quality and user utility	1
4	**Relevance feedback and XML retrieval (10)**	
4.1	Relevance feedback and pseudo relevance feedback	2
4.2	Evaluation of relevance feedback strategies	2
4.3	Global methods for query reformulation	2
4.4	Basic XML concepts, Challenges in XML retrieval	2
4.5	A vector space model for XML retrieval	1
4.6	Evaluation of XML retrieval	1
5	**Web search and Link analysis (10)**	
5.1	Web characteristics	1
5.2	The search user experience	1
5.3	Overview of Web crawling and indexes	2
5.4	The Web as a graph	2
5.5	Page Rank	1
5.6	Topic-specific Page Rank	2
5.7	Hubs and Authorities	1
	Total	**50**

Course Designer:

1.C.DEISY cdcse@ce.edu

Sub Code	Lectures	*Tutorial*	Practical	Credit
CCC	3	-	-	3

CCC - Data Mining Concepts and Techniques 3:0

Preamble: This course aims at facilitating the student to understand the concepts of data mining and various techniques involved in mining the data from the databases.

Prerequisite: C46 and C52

Program Outcomes addressed:

a. Graduates will demonstrate knowledge of mathematics science and engineering.
b. Graduate will demonstrate an ability to identify, formulate and solve engineering problems.
c. Graduates will demonstrate an ability to design and conduct experiments, as well as to analyze and interpret data.
I Graduates will show the understanding of impact of engineering solutions on the society and also will be aware contemporary issues.

Competencies

At the end of the course the student will be able to

1. Explain the concepts of Data mining and its applications.
2. Preprocess the given raw data for analysis using discretization and normalization.
3. Apply the mining techniques like association, classification and clustering on transactional databases.
4. Perform multi dimensional data mining using association and classification techniques.
5. Distinguish the types of databases like temporal, spatial and sequence.
6. Identify the type of analysis that can be done on the given databases.
7. Explain the concepts of spatial data mining, temporal data mining, text mining, web mining, visual data mining.
8. To perform mining on spatial, temporal and sequence databases.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10

2	Understand	40	20	30
3	Apply	40	60	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level learning objectives

Remember

1. Define Data Mining.
2. What do you mean by strong association rule?
3. How to select an attribute for classification?
4. What is cluster analysis?
5. Mention the purpose of correlation analysis
6. What do you mean by "stemming"? Give an example.

Understand

1. Compare and contrast the clustering and the classification techniques.
2. Can we do data mining on the data generated by the web? Justify
3. In what way "Over Fitting" can be avoided?
4. Illustrate the significance of candidate set generation step of level wise algorithm.
5. Describe the importance of pruning in decision tree construction with an example.
6. Given the two vector objects X=(1,1,0,0) and Y=(0,1,1,0) identify the similarity between these objects.

Apply

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. For the given database find all the frequent item sets using Apriori method and list all the strong association rules that match the metarule

$\forall$ x € transaction , buys(X,item1) ^ buys(X,item2) $\Rightarrow$ buys(X,item3).

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%

3. 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.

Name	Rank	Years	Turned
Mike	Assistant Professor	3	No
Mary	Assistant Professor	7	Yes
Bill	Professor	2	Yes
Jim	Associate Professor	7	Yes
Dave	Assistant Professor	6	No
Anne	Associate Professor	3	No

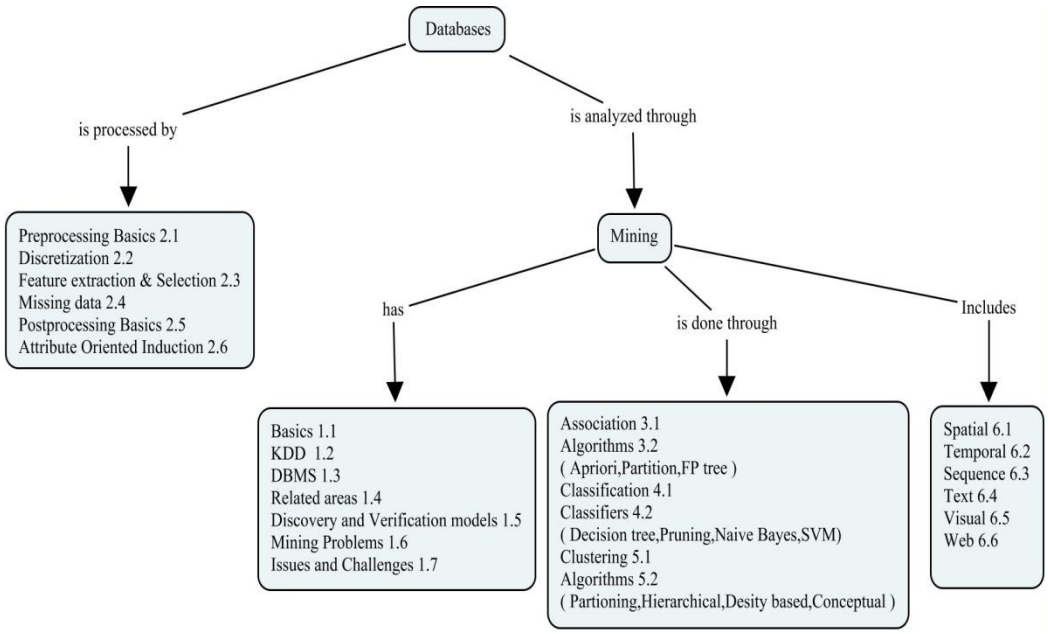
4. For the following Database, apply the entropy-based discretization for the numerical attribute and find the best split.

S.NO	Age	Credit rating
1.	25	Fair
2.	29	Excellent
3.	35	Fair
4.	42	Excellent
5.	47	Fair
6.	49	Excellent
7.	32	Fair
8.	34	Fair
9.	37	Excellent
10.	40	Fair
11.	44	Fair
12.	45	Excellent

5. Given two objects A1(22,1,42,10) and A2(20,0,36,8) compute the distance by Euclidean measure.
6. The data mining task wants to Cluster the following eight points (with (x,y) representing locations) into 3 clusters A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9). The distance function is Euclidean

distance. Initially assign A1, B1 and C1 as the center of each cluster respectively. Use K-Means algorithm to show the final three clusters.

Concept Map



Syllabus

Introduction to Data Mining – Concepts , KDD vs Data mining , DBMS vs Data mining , Other related areas , Discovery and Verification models , Other Mining Problems , Issues and Challenges, **Preprocessing –** Concepts , Discretization , Feature extraction & Selection , Missing data , Post processing , Attribute Oriented Induction, **Association Techniques -** Introduction Association Rules , Apriori algorithm , Partition algorithm , FP tree growth algorithm , Types of association rules, **Classification Techniques -** Introduction to Classification , Constructing decision tree – ID3 algorithm , Pruning , Naive Bayes Classification , SVM Classification, **Clustering Techniques -** Introduction to Clustering , Partitioning Method – K Means algorithm , Hierarchical Method – BRICH method , Density Based Method – DBSCAN method , Conceptual clustering – COBWEB algorithm, **Mining Applications** - Spatial data mining, Temporal data mining, Sequence mining, Text mining, Visual data mining, Web mining.

Text Books:

1. K.P.Soman, Shyam Diwakar, V.Ajay, "Insight into Data Mining – Theory and Practice", Prentice Hall of India, 2006. (Modules II, III and VI)

2. Arun K.Pujari, "Data Mining Techniques", Universities Press, 2001. (Modules I, IV, V and VI)

Reference Books:

1. Jiawei Han, Micheline Kamper, Data Mining: Concepts and Techniques Morgan Kaufman, 2000, ISBN: 1-55860-489-8. Chap1-3, 5-10.

2. M.H Dunham, "Data Mining: Introductory and advanced topics", Pearson Education, 2006.

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Introduction to Data Mining**	
1.1	Basics	1
1.2	KDD Vs. data mining	1
1.3	DBMS Vs. data mining	1
1.4	Related areas	1
1.5	Discovery and Verification models	1
1.6	Mining Problems	2
1.7	Issues and Challenges	1
2.	Processing	
2.1	Preprocessing Basics	1
2.2	Discretization	2
2.3	Feature extraction & Selection	1
2.4	Missing data	2

No.	Topic	No. of Lectures
2.5	Post processing Basics	0.5
2.6	Attribute Oriented Induction	0.5
3.	Association Techniques	
3.1	Introduction to Association Rules	1
3.2	Association Algorithms (Apriori, Partition, FP tree)	7
4	**Classification Techniques**	
4.1	Introduction to Classification	1
4.2	Classifiers (Decision tree, Pruning, Naive Bayes, SVM)	7
5	**Clustering Techniques**	
5.1	Introduction to Clustering	1
5.2	Algorithms (Partioning, Hierarchical, Desity based, Conceptual)	8
6	**Mining Applications**	
6.1	Spatial Data mining	1
6.2	Temporal Data mining	2
6.3	Sequence mining	3
6.4	Text mining	1
6.5	Visual Data mining	2

No.	Topic	No. of Lectures
6.6	Web mining	1
	Total	50

COURSE DESIGNERS:

1. Ms. A.M.Rajeswari amrcse@tce.edu
2. Ms. B.Subbulakshmi bscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCD	3	-	-	3

CCD Agent Based System 3 : 0

Preamble: This course introduces representations, techniques, and architectures used to build applied systems and to account for intelligence from a computational point of view.

Program me Outcomes addressed

d. An ability to identify, formulate and solve engineering problems

e. An ability to use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints

h. An ability to engage in life-long learning

Competencies

At the end of the course the student will be able to

1. Understand the behavior of agents including simple agent, goal-based agent, utility based agent, and reflex agent
2. Explain how agents can be used for informed and uninformed search, CSP, and adversial search.
3. Apply agent based searching to problems of traveling salesman, games, and scheduling.
4. Apply CSP and Adversarial searching to real time problems
5. Implement the Planning agents for Deterministic & Non Deterministic Domains
6. Apply learning methods including simulated annealing, genetic algorithms and local beam search to problems including vacuum world, path selection and mobile robot navigation for optimized results

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	10
2	Understand	30	30	30
3	Apply	50	50	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course level learning objectives

Remember

1. How to evaluate algorithm performance
2. Why we go for breadth first search
3. What is heuristic function
4. Define Plateaux
5. What is admissible heuristic in A*
6. What is Recursive Best First Search?
7. Define Critical path method in planning
8. What are the evaluation criteria for breadth first and depth first criteria

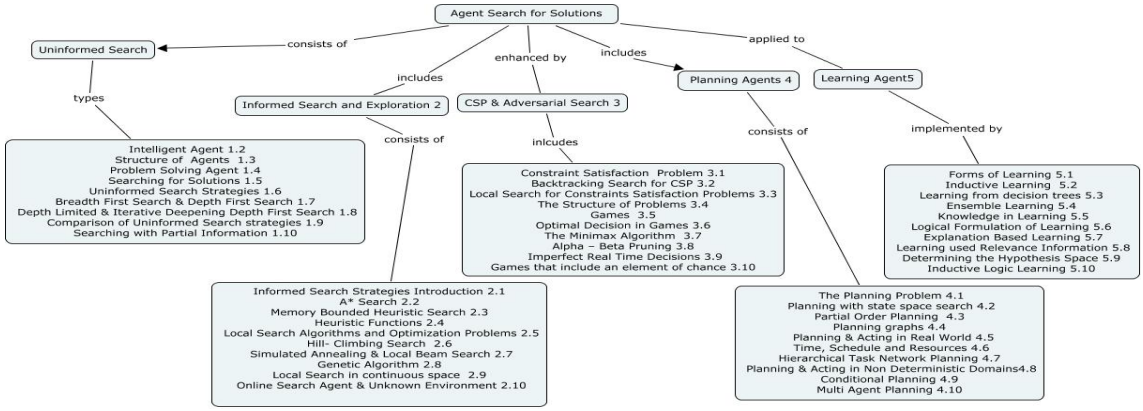
Understand

1. Explain Model based Reflex Agent
2. Explain the sequence used to define well defined problem
3. Explain Simulated Annealing Technique
4. Explain Games that include an element of chance
5. Explain Hierarchical Task Network Planning
6. Explain the Conditional planning in partially observable environment
7. How learning takes place in on-line search agent applied to robot navigation
8. How Minimax algorithm applied to two player games for optimal search?
9. How planning and Acting is carried in Nondeterministic Domains

Apply

1. Apply searching with partial information for the vacuum world problem assuming the dirt and machine in same room
2. Apply Genetic Algorithm steps involved for finding the mobile robot navigation path planning in an indoor environment
3. Brief about the role of Alpha Beta Pruning in Toy Games
4. Brief about the Multi Agent Planning applied for the Car Manufacturing industry in Generic level

Concept Map



Syllabus

Problem Solving Agents : Introduction about Artificial Intelligence, Intelligent Agent, Structure of Agents, Problem Solving Agents, Searching for solutions, Uninformed Search Strategies, Breadth First Search & Depth First Search, Depth Limited & Iterative Deepening Depth First Search, Comparison of Uninformed Search Strategies, Searching with partial information. **Informed Search and Exploration** : Informed Search and Exploration ,A* Search, Memory Bounded Heuristic Search, Heuristic Functions, Local Search Algorithms and Optimization Problems, Hill Climbing Search, Local Search in Continuous Spaces, Online Search Agents and Unknown Environments. **CSP and Adversarial Search**: Constraint Satisfaction Problems, Backtracking Search for CSP, Local Search for CSP, The Structure of Problem, Games, Optimal Decision Games, The Minimax Algorithm, Alpha – Beta Pruning, Imperfect Real Time Decisions, Games that include an element of chance. **Planning Agents**: The Planning Problem, Planning with State Space Search, Partial Order Planning, Planning Graphs, Planning and Acting in Real World, Time, Schedule and Resources, Hierarchical Task Network Planning, Planning & Acting in Non Deterministic Domains, Conditional Planning,Multi Agent Planning. **Learning Agents**: Forms of Learning, Inductive Learning, Learning from decision trees, Ensemble Learning, Knowledge in Learning, Explanation based Learning, Learning using Relevance Information, Determining the Hypothesis space, Inductive Logic Programming

Text Book:

1. Stuart Russell," Artificial Intelligence – A Modern Approach" Second Edition, Pearson Edition

References:

1. Elaine Rich, Kevin knight, Shivashankar B Nair, " Artificial Intelligence" Third Edition, Tata Mc Graw Hill

Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Problem Solving Agents (10)**	
1.6	Introduction about Artificial Intelligence	3
1.7	Intelligent Agent, Structure of Agents	1
1.8	Problem Solving Agent, Searching for solutions	1
1.9	Uninformed Search Strategies	1
1.10	Breadth First Search & Depth First Search	1
1.11	Depth Limited & Iterative Deepening Depth First Search	1
1.12	Comparison of Uninformed Search strategies	1
1.13	Searching with Partial Information	1
2.	**Informed Search and Exploration (10)**	
2.10	Informed Search Strategies Introduction	1
2.11	A* Search	1
2.12	Memory Bounded Heuristic Search	1
2.13	Heuristic Functions	1
2.14	Local Search Algorithms and Optimization Problems	1
2.15	Hill- Climbing Search	1
2.16	Simulated Annealing & Local Beam Search	1
2.17	Genetic Algorithm	1
2.18	Local Search in continuous space	1
2.19	Online Search Agent & Unknown Environment	1
3.	**CSP & Adversarial Search (10)**	
3.8	Constraint Satisfaction Problem	1
3.9	Backtracking Search for CSP	1

3.1	Local Search for Constraints Satisfaction Problems	1
3.1	The Structure of Problems	1
3.1	Games	1
3.1	Optimal Decision in Games	1
3.1	The Minimax Algorithm	1
3.1	Alpha – Beta Pruning	1
3.1	Imperfect Real Time Decisions	1
3.1	Games that include an element of chance	1
4.	**Planning Agents (10)**	
4.7	The Planning Problem	1
4.8	Planning with state space search	1
4.9	Partial Order Planning	1
4.10	Planning graphs	1
4.11	Planning & Acting in Real World	1
4.12	Time, Schedule and Resources	1
4.13	Hierarchical Task Network Planning	1
4.14	Planning & Acting in Non Deterministic Domains	1
4.15	Conditional Planning	1
4.16	Multi Agent Planning	1
5.	**Learning Agent (10)**	
5.8	Forms of Learning	1
5.9	Inductive Learning	1
5.10	Learning from decision trees	1
5.11	Ensemble Learning	1
5.12	Knowledge in Learning	1
5.13	Logical Formulation of Learning	1

5.14	Explanation Based Learning	1
5.15	Learning used Relevance Information	1
5.16	Determining the Hypothesis Space	1
5.17	Inductive Logic Learning	1
	Total	**50**

Course Designer:

1. D. Tamilselvi dtamilselvi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCE	3	-	-	3

CCE Wireless Networks 3:0

Preamble: The course aims at exploring the concepts of wireless networks, protocols, architectures and topologies.

Prerequisite: C45, C52

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

Competencies:

At the end of the course the student should be able to

1. Understand the propagation mechanisms including reflection, diffraction and scattering used in wireless communication
2. Understand wireless transmission techniques including short distance base band, UWB, pulse, carrier modulated and digital cellular
3. Understand the principles of cellular and Ad hoc networks and their standard (IEEE 802.11)
4. Understand topologies of cellular and ad hoc wireless network.
5. Understand mobility and power management in wireless networks.
6. Determine the performance of a given wireless network
7. Determine the features of a wireless network that meets a given set of specifications

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	30	20	20
2	Understand	50	50	40
3	Apply	20	30	40
4	Analyze	0	0	0

5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. What are the three important radio propagation phenomena at high frequencies?
2. What are the new elements added to the GSM infrastructure to support GPRS?
3. State the three types of services provided by GSM.
4. What is the difference between registration and call establishment?
5. What is location management?
6. What are the two steps in hand off?

Understand

1. Differentiate wireless and wired network.
2. Explain the concept involved in carrier modulated transmission.
3. Explain the topology followed in Bluetooth.
4. Explain the steps involved in generic handoff management process.
5. Distinguish between two types of independent mobile data networks.
6. Why is hexagonal cell shape preferred over square or triangular cell shape to represent the cellular architecture?
7. Give three reasons why it is difficult to detect collisions at the transmitter in wireless networks?
8. With necessary diagram explain IEEE 802.11 Architecture and Services.
9. Explain the different parts of location management adopted in GSM with example.
10. Explain the comparison of Ad hoc and infrastructure network topologies.

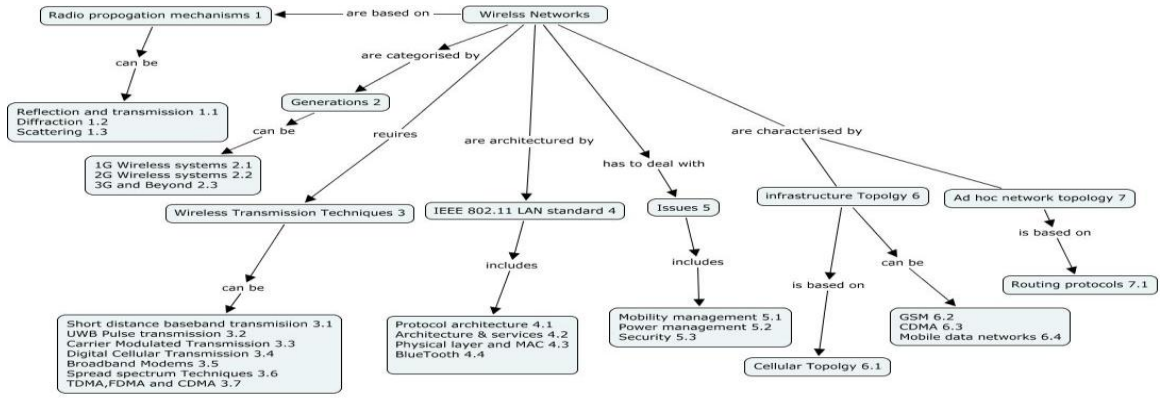
Apply

1. We want to transmit the binary sequence of data "10010110" by applying Manchester and Differential Manchester encoding on the bit pattern. Draw the original, Manchester and Differential Manchester encoded bit sequence.
2. In the following differential encoded Manchester signal
    a. Show the beginning and the end of each bit
    b. Identify all the bits in the data sequence
    c. Identify the bits if it was non-differential Manchester coded.

3. Using QPSK modulation and convolutional coding, the IS-95 digital cellular systems require 3 dB< S_r < 9 Db. The bandwidth of the channel is 1.25 MHZ and the transmission rate is R=9600 bps. Find the capacity of single IS-95 cell.
4. A vehicle travels at a cellular system at 100 kmph. How often will hand off occurs if the cell radius is i) 10 km ii) 5 km iii)500 m
5. If a signal interference ratio of 15dB is required for satisfactory forward channel performance of cellular system, what is the frequency reuse factor and cluster size that should be used for maximum capacity if the path loss exponent is 3? Assume that there are 6 co channel cells in the first tier and all of them are at same distance from mobile.
6. Determine the capacity for GSM with a carrier bandwidth of W_c =200 KHZ , number of users per carrier = 8 and Reuse factor K=3.
7. We want to provide a Radio communication service to a city. The total BW available is 30 MHZ,and each user requires 30 KHZ of BW for voice communication. If we employ a cellular topology where 30 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 compare to one antenna used.
8. We have installed cellular system with 100 sites, a frequency reuse factor of N=7, and 500 overall two-way channels:
    f) Give the number of channels per cell, total number of channels available to the service provider, and the minimum carrier-to-interference ratio (C/I)of the system in dB
    g) To expand the network, we decide to create an underlay-overlay system where the new system uses a frequency reuse factor of K=3. Give the

number of cells assigned to inner and outer cells to keep a uniform traffic density over the entire coverage area.

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 **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

References:

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 Lectures schedule

No	Topic	No of Lectures
1.	**Radio propagation mechanisms(1)**	
1.1	Reflection and transmission	0.5
1.2	Diffraction	0.25
1.3	Scattering	0.25
2	**Generations of wireless networks(2)**	
2.1	1G wireless systems	0.5
2.2	2G wireless systems	0.5
2.3	3G and beyond	1
3	**Wireless transmission techniques(8)**	
3.1	Short distance baseband transmission	1
3.2	UWB Pulse Transmission	1
3.3	Carrier Modulated transmission	1
3.4	Digital Cellular Transmission	1
3.5	Broad band modems for high speed	1
3.6	Spread spectrum techniques	1
3.7	Introduction to TDMA,FDMA and CDMA	2
4	**IEEE 802.11 LAN standard(11)**	
4.1	Protocol architecture	2
4.2	Architecture and services	2
4.3	Physical layer and MAC	2
4.4	Bluetooth overview	1

4.4.1	Radio specification	1
4.4.2	Base band specification	1
4.4.3	Link manager specification	1
4.4.4	Logical link control	1
5	**Issues(6)**	
5.1	Mobility management	2
5.2	Power management	2
5.3	Security	2
6	**Infrastructure Network Topology(20)**	
6.1	Cellular topology	2
6.1.1	Cell fundamentals	2
6.1.2	Signal to interference ratio calculation	1
6.1.3	Capacity expansion techniques	1
6.1.4	Network planning for CDMA systems	2
6.2	GSM	2
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	**50**

Course Designer:

1. C.Senthilkumar cskcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCF	3	-	-	3

CCF Grid Computing 3:0

Preamble: This course aims at facilitating students to understand the design issues and implementation details behind grid computing middleware. Students will learn an open set of grid standards and protocols that enable communication across heterogeneous, geographically dispersed environments.

Prerequisite: C55

Program Outcomes

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component, or process as per needs and specification.

Competencies:

At the end of the course, the students will be able to

1. Compare various grid computing techniques.
2. Understand various components of Grid architecture.
3. Understand the principles of merging grid services with web services.
4. Determine the various scheduling Techniques.
5. Provide Grid based solutions.
6. Understand the role of grid monitoring systems.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	20	20	20	20
2	Understand	30	20	30	20
3	Apply	50	60	50	60
4	Analyze	0	0	0	0
5	Evaluate	0	0	0	0
6	Create	0	0	0	0

Course level Learning Objectives

Remember

1. What are the software and Hardware requirements to create a Grid?

2. What are the basic goals of GGF?
3. List out types of Grid.
4. List out the various methods used for Scheduling.
5. Define the term services in SOA.
6. Give the role of Grid Resource broker.

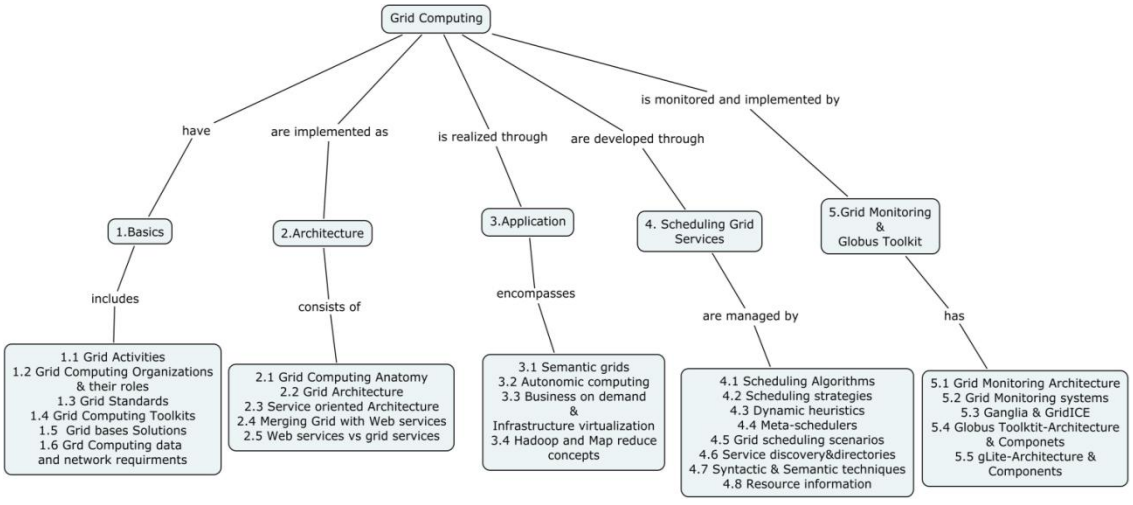
Understand

1. Explain the various Blocks in the architecture of a Grid.
2. Describe the service oriented architecture.
3. Discuss the role of a Meta Scheduler.
4. Compare Grid services with web services
5. Explain the various concepts involved in a Grid programming model.

Apply

1. Illustrate the layered Grid architecture with neat diagram.
2. A user would like to construct a grid for handling high throughput computing jobs which generally do not communicate with each other while running. Identify the grid computing toolkit to be used for this purpose and illustrate the features of the toolkit which you would use for implementing this grid application.
3. Demonstrate the Autonomic computing system with suitable examples.
4. Apply map reduce framework for word count problem which is to be run on top of Hadoop.
5. Construct a control flow graph for processing both batch and parallel jobs in a grid computing environment. Suggest a scheduling algorithm that suits to both cases.
6. Demonstrate Grid monitoring system with suitable example.

Concept Map



Syllabus:

BASICS: Grid activities -Grid computing organizations and their roles-Grid standards -Grid Computing Toolkits -Grid base solutions-Grid computing data & network requirements.

ARCHITECTURE: Grid computing anatomy –Grid Architecture - service oriented architecture- Merging the Grid with web services –Review on SOAP, XML & WSDL-Web services Vs Grid services.

APPLICATION: Semantic grid - Autonomic computing -Business on demand - infrastructure virtualization - Hadoop - Map Reducing concepts

SCHEDULING GRID SERVICES: Scheduling algorithms-Scheduling strategies: –Dynamic heuristics - Meta-schedulers - Grid scheduling scenarios– Service Discovery - Service directories - syntactic and semantic techniques- Resource information.

GRID MONITORING AND GLOBUS TOOLKIT: Grid Monitoring Architecture- Grid Monitoring Systems- Ganglia - GridICE- Globus toolkit-Architecture and Components-gLite-Architecture and Components

Text Books :

1. Joshy Joseph & Craig Fellenstein "Grid Computing", IBM Press, 2003.
2. Frederic Magoules, Thi-Mai-Huong Nguyen, Lei Yu, "Grid Resource Management: Toward Virtual and services Complaint Grid Computing", First Edition, CRC Press,2008. (Unit IV)
3. Barry Wilkinson, "Grid Computing: Techniques and Applications", CRC Press, 2010.

References:

1. Ian Foster & Carl Kesselman , "The Grid2 Blueprint for a New Computing infrastructure", Second Edition, Morgan Kaufmann, 2006.
2. Tom White, "Hadoop: The Definitive Guide MapReduce for the Cloud", O'Reilly media,First Edition,2009.
3. http://www.ibm.com/redbooks/

Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Basics(11)**	
1.1	Grid activities.	2
1.2	Grid computing organizations and their roles.	2

1.3	Grid standards.	1
1.4	Grid Computing Toolkits.	3
1.5	Grid base solutions.	1
1.6	Grid computing data and network requirements.	2
2.	**ARCHITECTURE (10)**	
2.1	Grid computing anatomy.	1
2.2	Grid Architecture.	2
2.3	Service oriented architecture.	3
2.4	Merging Grid with Web Services-Review on SOAP ,XML,WSDL	2
2.5	Web services Vs Grid services	2
3.	**APPLICATIONS(9)**	
3.1	Semantic grids.	2
3.2	Grid Computing Roadmap - Autonomic computing.	2
3.3	Business On Demand- Infrastructure Virtualization.	2
3.4	Hadoop and Map Reducing Concepts	3
4	**SCHEDULING GRID SERVICES: (10)**	
4.1	Scheduling algorithms.	1
4.2	Scheduling strategies.	1
4.3	Dynamic heuristics.	1
4.4	Meta-schedulers.	2
4.5	Grid scheduling scenarios.	1
4.6	Service Discovery, directories.	2
4.7	Syntactic and semantic Techniques.	1
4.8	Resource information.	1

5	**GRID MONITORING AND GLOBUS TOOLKIT (10)**	
5.1	Grid Monitoring Architecture	2
5.2	Grid Monitoring systems	1
5.3	Ganglia, GridICE	2
5.4	Globus Toolkit- Architecture & Components	3
5.5	gLite – Architecture & Components	2
	Total	**50**

Course Designer:

1. Dr. S. Padmavathi, spmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCG	3	0	-	3

CCG-Modeling and Simulation 3:0

Preamble

Modeling and Simulation refers analysis tool and design tool for various engineering system. Models can be constructed in mathematical ways to solve any network system. Random numbers can be generated and test the properties of random numbers by various method. In analysis of simulation, input values are generated and used for experiment. Output values are also analysised for a system. Finally the modeling and simulation concept is applied in CPU simulation, memory simulation

Prerequisite: Probability and Statistics, Queuing theory

Program Outcomes Addressed

a. Graduates will demonstrate knowledge of Mathematics Science and Engineering.

b. Graduates will show an ability to identify, model and solve any engineering problems.

c. Graduate will develop confidence for self education and ability for life-long learning.

Competencies

At the end of the course, the student will be able to

1. Solve any network problem using queuing theory.
2. Generate random numbers for simulation experiment
3. Test the properties of random numbers
4. Produce the input data to simulation experiment using random numbers.
5. Test the input data
6. Model the CPU and memory unit in computer system

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3/End-semester examination**
1	Remember	10	10	0
2	Understand	30	30	30
3	Apply	60	60	70

4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. What is discrete simulation?
2. What is verification?
3. Define convolution
4. Different type of computer simulation
5. What is goodness of fit?
6. What is terminating simulation?

Understand

1. Write the application of simulation
2. What is CPU simulation?
3. Write the stochastic nature of output data
4. What is run length with example?
5. Write Uniform random variate on (a,b)
6. How do you collect the data for simulation experiment**?**

Apply

1. Records pertaining to the monthly number of job- related injuries at an underground coal mine being studied by a federal agency. The values of the past 100 months were as follows.

Injuries per month	Frequency of occurrence
0	35
1	40
2	13
3	6
4	4
5	1
6	1

Apply the chi-square test to these data to test the hypothesis that the underlying distribution is Poisson. Use a level of significance of $\alpha = 0.05$

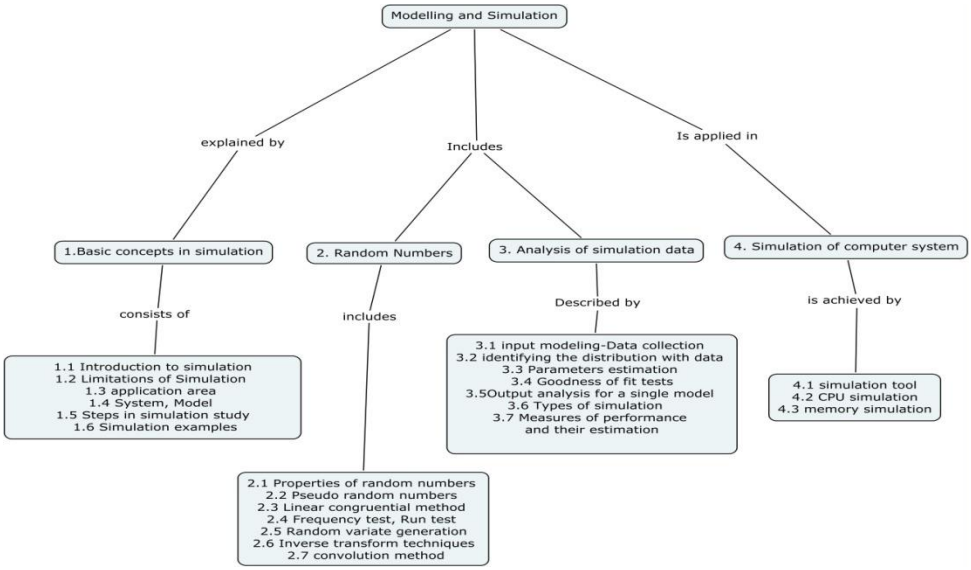
2. Use chi-square test with $\alpha = 0.05$ to test whether the data shown below are uniformly distributed.

0.34 0.90 0.25 0.89 0.87 0.44 0.12 0.21 0.46 0.67 0.83 0.76 0.79 0.64 0.70
0.81 0.94 0.74 0.22 0.74 0.96 0.99 0.77 0.67 0.56 0.41 0.52 0.73 0.99 0.02
0.74 0.30 0.17 0.82 0.56 0.05 0.45 0.31 0.78 0.05 0.79 0.71 0.23 0.19
0.82 0.93 0.65 0.37 0.39 0.42 0.99 0.17 0.99 0.46 0.05 0.66 0.10 0.42
0.18 0.49 0.37 0.51 0.54 0.01 0.81 0.28 0.69 0.34 0.75 0.49 0.72 0.43
0.56 0.97 0.30 0.94 0.96 0.58 0.73 0.05 0.06 0.39 0.84 0.24 0.40 0.64
0.40 0.19 0.79 0.62 0.18 0.26 0.97 0.88 0.64 0.47 0.60 0.11 0.29 0.78

3. Based on runs up and down, determine whether the following sequence of 40 numbers is such that the hypothesis of independence can be rejected where $\alpha = 0.05$ 0.41 0.68 0.89 0.94 0.74 0.91 0.55 0.62 0.36 0.27 0.19 0.72 0.75 0.08 0.54 0.02 0.01 0.36 0.16 0.28 0.18 0.01 0.95 0.69 0.18 0.47 0.23 0.32 0.82 0.53 0.31 0.42 0.73 0.04 0.83 0.45 0.13 0.57 0.63 0.29

4. Generate discrete uniform variate with pmf $p(x) = \frac{2x}{k(k+1)}$, x = 1, 2,3 ..k
5. Applying simulation concepts, how do you design the CPU?
6. Construct memory unit using simulation

Concept Map



Syllabus

Basic concepts in simulation: Introduction to simulation, Limitations of Simulation, application area, System, system environment, components, Model, types of model, Steps in simulation study, Simulation examples- queuing system. **Random Numbers:** Properties of random numbers, Pseudo random numbers, linear congruential method, Frequency test, Run test, random variate generation-Inverse transform techniques, convolution method **Analysis of simulation data**: input modeling-Data collection, identifying the distribution with data, Parameters estimation, Goodness of fit tests. Output analysis for a single model- Types of simulation with respect to output analysis, stochastic nature of output data, Measures of performance and their estimation, Output analysis for terminating simulations **Simulation of computer system**: Introduction, simulation tool, CPU simulation, memory simulation

Text Book:

1. Jerry Bank, John S. Carson II, Barry L. Nelson, David M. Nicol, " Discrete Event System Simulation ", Prentice Hall of India, New Delhi, 2003

References:

1. Narasingh Deo, "System Simulation with Digital Computer", PHI, New Delhi, 1997
2. Averill M.Law, W.David Kelton," Simulation Modeling and Analysis", McGraw Hill, New Delhi, 2000

Course contents and Lecture schedule

S.No.	Topics	No. of Lectures
1.	**Basic concepts in simulation**	
1.1	Introduction to simulation	1
1.2	Limitations of Simulation, application area	1
1.3	System, system environment, components	1
1.4	Model, types of model	1
1.5	Steps in simulation study	2
1.6.	Simulation examples-queuing system	4
2	**Random Numbers**	

2.1	Properties of random numbers, Pseudo random numbers	1
2.2	Linear congruential method	1
2.3	Frequency test	2
2.4	Run test	3
2.5	Inverse transform techniques	5
2.6	Convolution method	2
3	**Analysis of simulation data**	
3.1	Introduction, Data collection	1
3.2	Identifying distribution	1
3.3	Parameters estimation	4
3.4	Goodness of fit tests	3
3.5	Types of simulation with respect to output analysis	1
3.6	Stochastic nature of output data	1
3.7	Measures of performance and their estimation	1
3.8	Point estimation	2
3.9	Interval estimation	2
3.10	Output analysis for terminating simulations	3
4	**Simulation of computer system**	
4.1	Introduction	1
4.2	simulation tool	2
4.3	CPU simulation	2
4.4	memory simulation	2

Course Designers:

1. V.Mohan vmohan@tce.edu
2. M.Mutharasan mmmat@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCH	2	1	-	3

CCH Parallel Computing 3:0

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: C23- Computer Organization and Design and C35-Design and Analysis of Algorithms

Programme Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve highly computational problems.

c. Graduates will demonstrate an ability to design, analyze and interpret data.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

Competencies

At the end of this course, the student will be able to:

1. Understand the basics of parallel processing terminology and theoretical model of parallel computation.
2. Understand the fundamentals of parallel programming platforms, principles of design and parallel algorithm models.
3. Use mapping and scheduling techniques for parallel application in cluster computing environment.
4. Use Domain Decomposition techniques and Parallel algorithm models to design a parallel algorithm for computationally intensive applications like Dictionary operations and Fast Fourier transform.
5. Design and analyze the parallel implementation of matrix multiplication for n x n matrix size and sorting algorithms
6. Apply parallel approach for computationally intensive applications like combinatorial search and dictionary operations.

7. Analyze the complexities involved in searching an optimal solution for intelligent games such as chess, checkerboard games.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	10
2	Understand	20	20	10
3	Apply	30	30	40
4	Analyze	30	30	40
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember:

1. What is Control Pararllelism?
2. What is meant by pipelining?
3. List out the PRAM Variants.
4. Define: Brent's theorem
5. What is Hyperquicksort?
6. Define: Alpha-Beta Pruning?

Understand:

1. Differentiate Binary k-cube from a cube-connected network of degree k.
2. When a graph is said to be cubical?
3. Differentiate Multiprocessor from Multicomputer.
4. How to map a 8 node ring into a hypercube?
5. Write the disadvantages in Stone's Bitonic sort.
6. How to overcome the complexity in parallel search?

Apply:

1. Draw the 8-Processor de-Brujin Network.
2. Devise a PRAM algorithm for finding the sum of 'n' numbers using n / 2 processors.
3. Implement the Hyperquick sort on the following sequence :

   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}

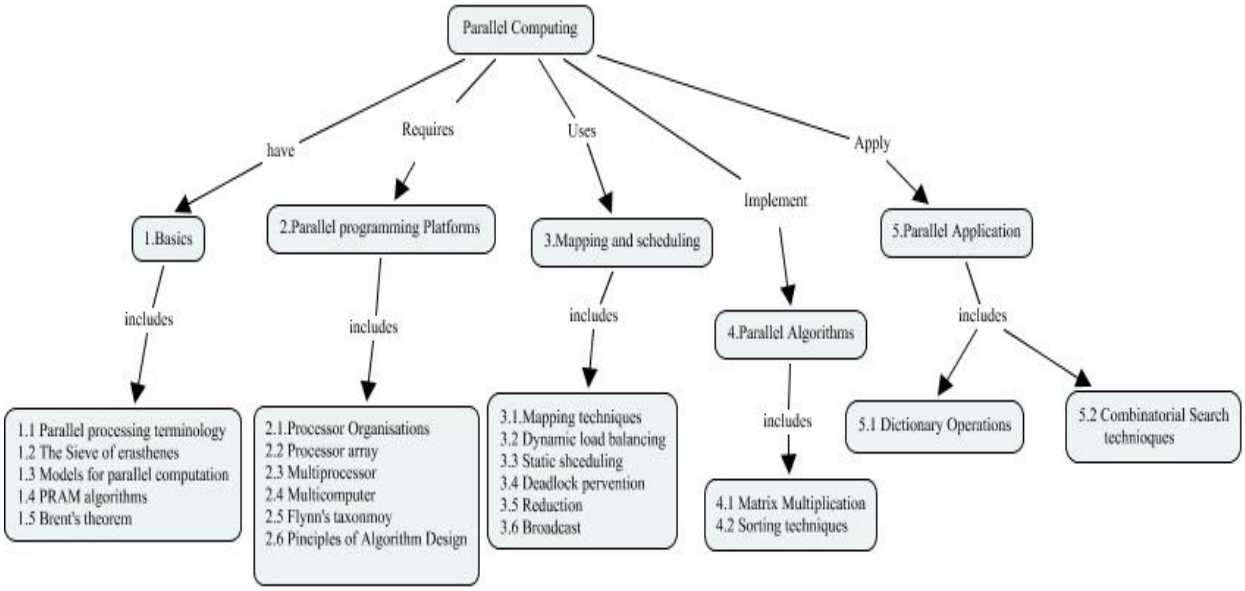
No.of.Processors = 4

4. Design the two dimensional mesh and 8-processor Shuffle-Exchange network.
5. Design Bitonic Merge sort using Stone's perfect shuffle interchange.
6. Design a Matrix multiplication algorithm for Shuffle Exchange network.

Analyze:

1. Generate the sequence of prime numbers using the Sieve of Erasthenes based on data-parallel approach with 'n' processors and analyze its time complexity.
2. Discuss Parallel Quicksort algorithm for UMA multiprocessors for a set of 32 numbers.
3. Identify an efficient algorithm for searching operations in a Dictionary and evaluate its time complexity.
4. Analyze the parallel matrix multiplication algorithm for 2-D Mesh network with 16 processors and find its time complexity.
5. Analyze the complexity and determine the number of processors required for the Prefix summation for a set of 16 elements.
6. Analyze the alpha beta search algorithm and evaluate the complexities in subtree generation for chess game tree.

Concept Map



Syllabus

Basics: Introduction, Parallel processing terminology, The Sieve of Erasthenes, 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 Books:

1.Michael J. Quinn, "Parallel Computing –Theory and Practice", Tata McGraw Hill Publishing Company Limited, Second edition, 2006.

Reference Book:

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 Content and Lecture Schedule

S.No.	Topic	No of Lectures
1.	**Basics**	
1.1	Introduction: Parallel Processing terminology	1
1.2	The sieve of Erasthenes	1

S.No.	Topic	No of Lectures
1.3	Models for parallel computation	1
1.4	PRAM Algorithms	2
1.5	Brent's Theorem	1
2.	**Parallel Programming – Platforms, Principles and Models**	
2.1	Processor Organizations	1
2.2	Processor arrays, Multiprocessor	2
2.3	Multicomputers, Flynn's Taxonomy	1
2.4	Principles of Parallel Algorithm Design	1
2.5.1	Decomposition Techniques	1
2.5.2	Characteristics of tasks and Interactions	1
2.5.3	Mapping Techniques for Load Balancing	1
2.6	Parallel Algorithm Models	1
3.	**Mapping and scheduling**	
3.1	Mapping data to processors on processor arrays and multicomputers	2
3.2	Dynamic load balancing on multicomputers	1
3.3	Static scheduling on UMA multiprocessor	2
3.4	Deadlock	1
3.5	Reduction	2
3.6	Broadcast	1
3.7	Prefix sums	1
4	**Parallel Algorithms**	
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	3
5	**Parallel Applications**	
5.1	Dictionary operations-Complexity of parallel search	2
5.1.1	Searching on multiprocessors	1

S.No.	Topic	No of Lectures
5.2	Combinatorial search-Divide and Conquer	1
5.2.1	Branch and Bound	2
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 Periods	**45**

Course Designer:

1. S.Padmavathi spmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCJ	3	0	-	3

CCJ Service-Oriented Architecture 3:0

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. 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

- C55: Web Programming, C54: Computer Networks: Principles
- C42: Computer Networks: Use & Configuration

Program Outcomes addressed

b. Graduate will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and Specification.

f. Graduates will demonstrate skills to use modern engineering tools, software's and equipments to analyze problems.

j. Graduate will develop confidence for self education and ability to engage in life-long learning.

Competencies

At the end of the course, the student will be able to

1. Describe the basics of SOA.
2. Explains how vendors and standard organizations have formed a competitive and collaborative arena.
3. Describe the underlying paradigm primarily responsible for defining SOA.
4. Distinguish SOA architectural model from other architectures.
5. Explain the robust, coordinated, and transaction-capable service-oriented architecture.
6. Explain the use of the WS-BPEL language to create a business process definition.
7. Illustrate the current technology supporting primitive and contemporary SOA models.
8. Apply the SOA design principles in all type of application development to develop web services.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	10	20
2	Understand	20	20	20
3	Apply	10	20	60
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. What is Contemporary SOA?
2. List down the benefits of SOA.
3. What is the role of Service Requestor?
4. Give the basic structure of SOAP message?
5. Show how SOA is re-shaping XML and Web services?
6. Write about the basic components that defines SOA as an architecture model?

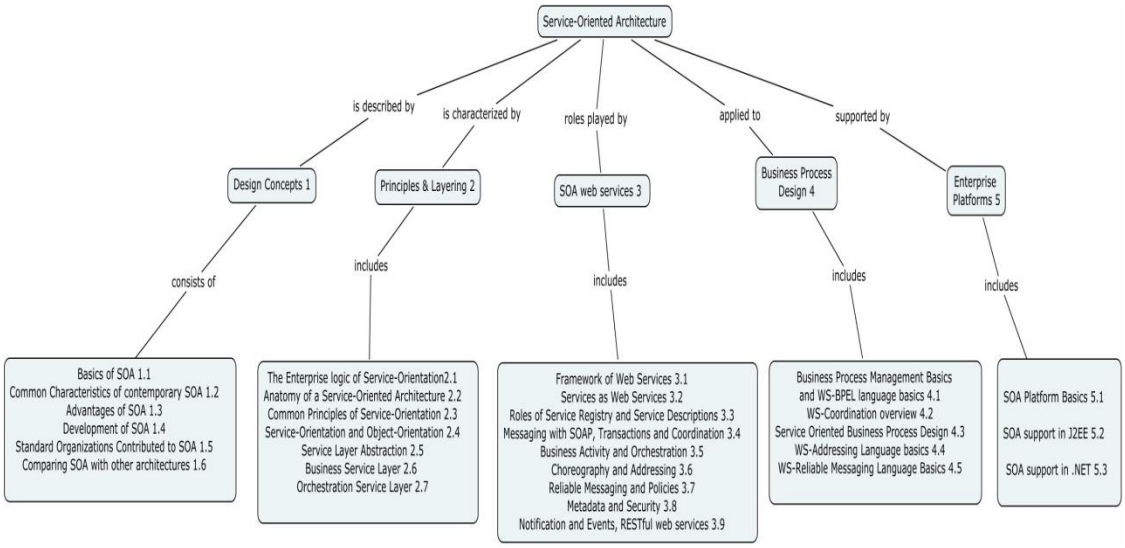
Understand

1. Draw the basic Structure of SOAP message?
2. Discuss about Application architecture, Enterprise architecture and Service-Oriented architecture?
3. Explain Service-Oriented Architecture and Client-Server architecture?
4. Describe about the various Service models?
5. Describe about the WS-Policy Framework?
6. Explain about the Service Provider and Service Requestor in J2EE platform?

Apply

1. Demonstrate the Principles of Service-Orientation?
2. Illustrate with neat diagrams the fundamental roles played by a Service?
3. Summarize the principles behind Service-Orientation with Object-Orientation?
4. Illustrate the primary characteristics of Contemporary SOA?
5. Illustrate the use of WS-Addressing Specification in SOA?
6. Illustrate the Step-by-step design of WS-BPEL Process Definition?

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:

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**	
1.1	Fundamental SOA	1
1.2	Common Characteristics of contemporary SOA	2
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	2
2	**Principles of Service-Orientation and Service Layer**	
2.1	Services-Orientation and The Enterprise	1
2.2	Anatomy of a Service-Oriented Architecture	2
2.3	Common Principles of Service-Orientation	1
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**	
3.1	The Web Services Framework - Overview	1

S.No.	Topic	No. of Lectures
3.2	Services as Web Services	1
3.3	Service Registry and Service Descriptions	2
3.4	Messaging with SOAP, Transactions and Coordination	2
3.5	Business Activity and Orchestration	2
3.6	Choreography and Addressing	2
3.7	Reliable Messaging and Policies	1
3.8	Metadata and Security	1
3.9	Notification and Events, RESTful Web services: The basics	2
4	**Business Process Design**	
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	2
4.5	WS-Reliable Messaging Language Basics	2
5	**Enterprise Platforms and SOA**	
5.1	SOA Platform Basics	2
5.2	SOA support in J2EE	2
5.3	SOA support in .NET	2
	Total	**45**

Course Designer:

1. G. Madhu Priya gmadhupriya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCK	3	0	-	3

CCK Software Risk Assessment and Management 3:0

Preamble: Risk Assessment and Management describe the key elements necessary to have a successful software risk management program and provide at least one strategy for overcoming each type of risk.

Prerequisite: C45 – Software Engineering, C56 – Software Design, C63 – Software Verification & Validation

Programme Outcomes addressed

b. Graduates will identify, formulate and solve risks in engineering problems.

e. Graduates will use techniques, skills, and modern tools to identify, manage and mitigate risks in software projects.

h. Graduates will visualize and work on real time scenario to reduce risks and its impact.

Competencies

At the end of the course the students will be able to

1. Understand the risk culture and to manage them with the help of models.
2. Identify risk using different methods including history based methods and intuitive methods.
3. Identify risk in various levels including process, project enterprise and product.
4. Respond to the identified risks through risk avoidance, risk transfer, risk acceptance, risk monitoring and risk mitigation.
5. Track the various kinds of risks like project and operational risks with the use of models including matrix model, tree model, Failure mode effects analysis (FMEA), Affinity diagrams, Risk lines, Probability Density Function (PDF) and risk simulation.
6. Manage the risks through hierarchical holographic modeling and theory of scenario of structuring.
7. Perform Decision analysis through various methods like decision rules, decision trees, decision Matrix, Fractile Method, Triangular distribution and influence diagrams.
8. Perform multi-objective trade –off analysis using Surrogate worth trade-off (SWT) method and Utility Function Approach.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	20
2	Understand	20	20	20
3	Apply	40	40	40
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	20	20	20

Learning Objectives

Remember

1. What is a Risk?
2. How is goal setting done in risk driven projects?
3. What are the models of Risk management?
4. Differentiate external with internal risks?
5. What are 3 P's?
6. What is out of box thinking?

Understand

1. Discuss the origin of risks?
2. Explain how decision trees are useful in analyzing risks.
3. Discuss the fractile method for decision analysis?
4. Discuss the W method for test generation.
5. Explain the various ways of performing decision analysis?
6. Discuss the surrogate worth trade-off method for multi-objective trade-off analysis?

Apply

1. For an international football match, identify the Risks that could be encountered based on software using any three identification methods.
2. For the above scenario, perform risk distribution analysis method and generate an analysis report.
3. Explain the various responses that could be performed for the above risks.

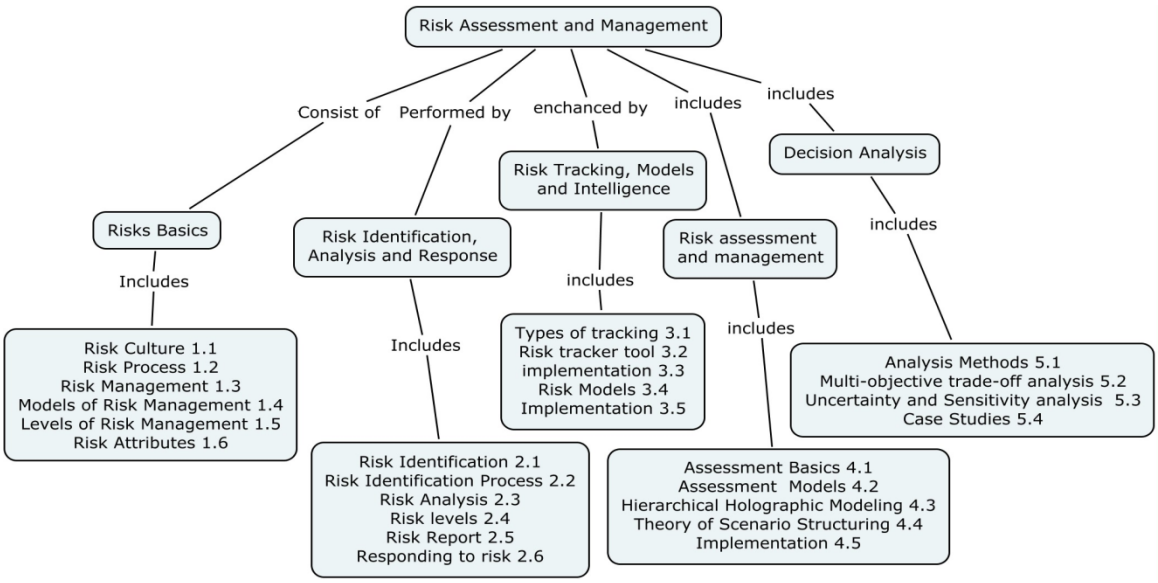
Create

1. Perform a Failure Mode Effect Analysis (FMEA) for the various risks identified in a multi-client software project of booking tickets.

2. Compute the various risk metrics for the above scenario.
3. Construct a Affinity diagram and a tree model for a software project that performs ERP for an organization.
4. The department of public works for a large city decided to develop a Web-based Pothole tracking and repair system. The description is as follows:

Citizens can log onto a web site and report the location and severity of potholes. As potholes are reported they are logged within a "public works department repair system" and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb etc.) district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and includes pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and dollar amount of damage. Identify the risks, and apply any two risk tracking methods and generate reports. Have your own assumptions of time duration.

Concept Map



Syllabus

Software Risks: Culture – What is risk, Risk Driven Project management, Controlling the Process, Environment and risk, Maturity in risk culture, Risk scale,

Preparing for risk. Risk Management Process - Five Models of Risk Management, Risk management at different levels, Risk Escalation. Risk attributes – Classification, Origin, Screening, Three P's, Severity, Levels, affected process areas, affected key result areas, goals, and requirements; **Risk Identification, Analysis and Response:** Risk Identification- Methods, Levels, Implementing Risk Identification Process. Risk Analysis- Scope and Purpose, First order analysis, Risk Distribution analysis, Larger Picture of analysis, Risk levels and analysis effort, Analysis report. Responding to risk – Treatment for Catastrophic risks, constraint risks, responding to ordinary threats, Risk response plans, risk avoidance, transfer, acceptance, Monitoring, Mitigation, Contingency and Strategic plan; **Risk Tracking, Models and Intelligence:** Types of tracking, trigger levels, tracking project risks, Operational risks, Enterprise risks, learning by tracking, risk tracker tool, hardening of risks, implementation. Risk Models- Implementation. Intelligence- metrics Models, Earned value model, estimation model, requirement model, critical path Model, WBS Model, PERT Model, implementation; **Risk assessment and management:** Assessment Process, Information, intelligence and Models, building blocks of mathematical models. Identifying risk through Hierarchical Holographic modeling – Hierarchical Aspects, Hierarchical Overlapping Co-ordination, Hierarchical Holographic Modeling(HHM) , Theory of Scenario Structuring, HHM in System acquisition Project, Software acquisition; **Decision Analysis:** Decision rules under uncertainty, Decision trees, Decision Matrix, Fractile Method, triangular distribution, Influence diagrams. Multi-objective trade-off analysis - Surrogate Worth trade-off (SWT) Method, Utility function approach. Uncertainty and Sensitivity analysis - Basics, Uncertainty Sensitivity Index Method (USIM), Risk Filtering and ranking and management, case studies.

Text Books :

1. C. Ravindranath Pandian, "Applied Software risk Management - A Guide for Software Project Managers", First edition, 2007.

2. Yacovy.Haimes, "Risk Modeling, Assessment and Management", Second Edition, A John Wiley & Sons, Inc., Publication, 2004.

References:

1. The Essentials of Risk Management: The Definitive Guide for the Non-risk Professional by Michel Crouhy, Dan Galai and Robert Mark, RSC Publishing, 2001.

Course Content and Lecture Schedule

S.No.	Topic	No of Lectures
1.	**Software Risks**	
1.4	Risk culture	2
1.5	Risk Process	1
1.6	Risk Management	2
1.7	Models of Risk Management	2
1.8	Levels of Risk Management	2
1.9	Risk Attributes	1
2.	**Risk Identification, Analysis and Response**	
2.4	Risk Identification	2
2.5	Risk Identification Process	1
2.6	Risk Analysis	2
2.7	Risk levels	2
2.8	Risk Report	1
2.9	Responding to risks	2
3.	**Risk Tracking, Models and Intelligence**	
3.8	Types of Tracking	2
3.9	Risk Tracker Tool	2
3.10	Tracking Implementation	3
3.11	Risk Models	2
3.12	Implementation	1
4	**Risk assessment and management**	
4.3	Assessment Basics	2
4.4	Assessment Models	3
4.5	Hierarchical Holographic Modeling(HHM)	2
4.6	Theory of Scenario Structuring	2
4.7	Implementation	1
5	**Decision Analysis**	
5.3	Analysis Methods	2
5.4	Multi-objective trade-off analysis	2
5.5	Uncertainty and Sensitivity Analysis	4
5.6	Case Studies	2
	Total	**50**

Course Designer:

1. A. Askarunisa aacse@tce.edu

Sub Code	Lecture	Tutorial	Practical	Credit
CCL	3	--	--	3

CCL Data warehousing 3:0

Preamble:

This course aims at facilitating the student to understand the various functionalities, ability to quickly analyze large and multidimensional data sets of a data warehouse. Data warehouse cleanses and organizes data to allow users to make business decisions based on facts. Enterprises have realized the importance of data warehousing solutions in making strategic business decisions, and have either implemented or planning to implement data warehouses.This course provides a practical line for planning, building and managing a successful data warehouse.

Prerequisite: C46 Database practice

C52 Database principles and design

Program outcomes addressed

a. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, mathematics and science

c. Graduates will demonstrate an ability to design a system or component or process to meet stated specifications

e. Graduates will demonstrate an ability to use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints

g. Graduates will demonstrate an ability to function on multidisciplinary teams

i. Graduates will demonstrate an ability to consider social, environmental, economic and ethical impact of engineering activities in a given context.

j. Graduates will demonstrate an ability to consider issues from global and multilateral views.

Competencies:

At the end of the course, the student will be able to

1. Understand the life cycle approach for a data warehouse project.
2. Design the architectural components of the data warehouse.
3. Design Logical and physical model of the data warehouse.
4. Perform the data extraction, transformation, and loading (ETL) functions.
5. Deploy and maintain the data warehouse.
6. Analyze large multidimensional data
7. Create relations for the datawarehouse

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	40	30	20
2	Understand	40	30	30
3	Apply	20	20	20
4	Analyze	0	10	10
5	Evaluate	0	0	0
6	Create	0	10	20

Course Level Learning Objectives

Remember

1. Name at least five characteristics or features of a data warehouse.
2. List the various data sources for the data warehouse.
3. What is the STAR schema?
4. What are aggregate fact tables?
5. Name four key issues to be considered while planning for a data warehouse.
6. What is meant by conforming dimensions?

Understand

1. A data warehouse is an environment, not a product, Discuss.
2. What types of processing take place in a data warehouse? Describe.
3. How are the top-down and bottom-up approaches different for building a data warehouse? Discuss the merits and disadvantages of each approach.
4. How are data warehouse projects different from OLTP system projects?
5. Explain any four development phases in the life cycle of a data warehouse project.
6. What is your understanding of data warehouse architecture? Describe

Apply

1. You are the manager for the data warehouse project of a nationwide car rental company. Your data warehouse is expected to start out in the 500 GB range. Select the suitable server hardware and write a justification for that.
2. In a STAR schema to track the shipments for a distributions company, the following dimension tables are found: (1) time, (2) customer ship-to, (3) ship-from, (4) product, (5) type of deal, and (6) model of shipment. Review these dimensions and list the possible attributes for each of the dimension tables.

3. As the data warehouse administrator, performance enhancement is high on your list. Highlight the techniques you plan to adopt. For each technique, indicate tasks necessary to implement the technique.

4. Assume you are in the insurance business. Find two examples of Type 2 slowly changing dimensions in that business. Write the specifications for applying the Type 2 changes to the data warehouse with regard to the two examples.

5. You are the data design specialist on the data warehouse project team for a retail company. Apply STAR schema to track the sales units and sales dollars with three dimension tables. Explain how you will decide to select and build four two-way aggregates.

6. Apply a simple STAR schema with a factless fact table to track patients in a hospital by diagnostic procedures and time.

Analyze

1."In a data warehouse, business requirements of the users form the single and most powerful driving force". Do you agree? If you do, state four reasons. If not, is there any other such driving force?

2. Give three reasons why you think ETL functions are most challenging in a data warehouse environment.

3. Compare and Contrast MOLAP and ROLAP for an insurance company's data warehouse.

4. Analyze the reasons for backing up a data warehouse. How is this different from backing up in an OLTP system?

5. Do you think a multitier user support structure is suitable for the data warehouse environment? Analyze the alternates.

6. Analyze how physical model components are related to components of the logical model.

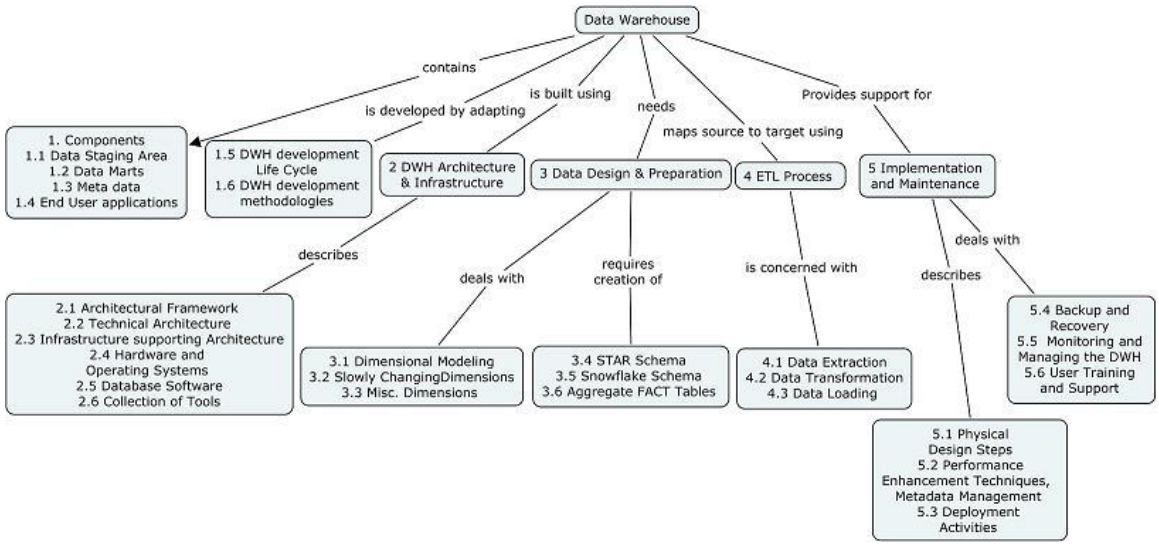
Create

1. You are the data design specialist on the data warehouse project team for a manufacturing company. Design a STAR schema to track the production quantities. Production quantities are normally analyzed along the business dimensions of product, time, parts used, production facility, and production run.
2. In your organization, assume that customer names and addresses are maintained in three customer files supporting three different source operational systems. Describe the possible entity identification problem you are likely to face when you

consolidate the customer records from the three files. Write a procedure outlining how you propose to resolve the problem.

3. As the data warehouse administrator, prepare a backup and recovery plan. Indicate the backup methods and schedules. Explore the recovery options. Describe the scope of the backup function. How will you ensure the readiness to recover from disasters?
4. As the manager for the data warehouse project, write a project completion report to your CIO and the executive project sponsor. List the major activities completed. Mention the plan for staged deployment of future releases. Indicate the plans for ongoing maintenance. Briefly highlight each topic on growth and maintenance.
5. Your project team has decided to use the system logs for capturing the updates from the source operational systems. You have to extract data for the incremental loads from four operational systems all running on relational databases. These are four types of sales applications. You need data to update the sales data in the data warehouse. Make assumptions and describe the data extraction process.
6. As the data designer for an international bank, consider the possible types of snapshot and transaction tables. Complete the design with one set of snapshot and transaction tables.

Concept Map



Syllabus

Components of data warehouse: Data staging area, Data Marts, Metadata, End user applications, DW development life cycle, DW development Methodologies **Data Warehouse Architecture and Infrastructure:** Understanding DWH Architecture, Architectural Framework, Technical Architecture, Infrastructure Supporting Architecture, Hardware and Operating Systems, Database Software, Collection of Tools **Data Design and Preparation:** Dimensional Modeling, Slowly Changing Dimensions, Miscellaneous Dimensions, STAR Schema, Snowflake Schema, Aggregate Fact Tables **ETL process:** Data Extraction, Data Transformation, Data Loading **Implementation and Maintenance of DWH:** Physical Design Steps, Performance Enhancement Techniques, Meta Data Management, Deployment Activities,. Backup and Recovery, Monitoring and managing the Data warehouse, User Training and Support

Text Books:

1. Data warehousing fundamentals Ponnaiah, Paulraj. Wiley Publishers 2003.
2. The Data warehouse Life cycle tool kit, 2nd edition Kimball, Ralph. Wiley Publishers 2008.

References:

1.Data Warehousing: Design, Development and best practices Mohanty, Soumendra. Tata McGraw Hill 2006.

2.Data warehousing in the real world Anahory, Sam / Murray, Dennis. Addison Wesley publishers 2003.

Websites

1.http://www.inmoncif.com

2.http://www.ralphkimball.com

3.http://www.datawarehousing.com

Course Contents and Lecture Schedule

S.No.	Topic	No. of Lectures
1	**Components of data warehouse**	
1.1	Data Staging Area	1
1.2	Data Marts	1
1.3	Metadata	1
1.4	End User Applications	1

S.No.	Topic	No. of Lectures
1.5	DWH development life cycle	2
1.6	DWH development Methodologies	2
2	**Data Warehouse Architecture and Infrastructure**	
2.1	Architectural Framework	2
2.2	Technical Architecture	2
2.3	Infrastructure Supporting Architecture	1
2.4	Hardware and Operating Systems	1
2.5	Database Software	1
2.6	Collection of Tools	1
3	**Data Design and Preparation**	
3.1	Dimensional Modeling	2
3.2	Slowly Changing Dimensions	1
3.3	Miscellaneous Dimensions	1
3.4	STAR Schema	2
3.5	Snowflake Schema	1
3.6	Aggregate Fact Tables	1
4	**ETL Process**	
4.1	Data Extraction	3
4.2	Data Transformation	3
4.3	Data Loading	2
5	**Implementation and Maintenance of DWH**	
5.1	Physical Design Steps	2
5.2	Performance Enhancement Techniques, Meta Data Management	2
5.3	Deployment Activities	1
5.4	Backup and Recovery	1
5.5	Monitoring and managing the Data warehouse	1
5.6	User Training and Support	1

Course Designer:

1. M.Sivakumar mskcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCM	3	0	-	3

CCM Enterprise Project Development using FOSS 3:0

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 C34-Operating Systems: Configuration and Use, C44**-**Operating Systems: Principles and Design

Programme Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

e. Graduates will use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

k. Graduate who can participate and succeed in competitive examinations.

Competencies

At the end of the course, the student will be able to

1. Explain GNU/Linux Architecture and Linux Distributions
2. How to Build OS from source and Use Libraries and packages
3. Explain the File Handling and Graphics Tools
4. Create Linux Web Server Using LAMP
5. Apply String access mechanisms and Sequential and Random access methods to Online Dictionary
6. Apply the Shells and Scripting Languages to calculate the Prime numbers in a given range
7. Compare the Copyright assignment and Ownership and Dual Licensing Schemes

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	20
2	Understand	30	40	30
3	Apply	40	40	40
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	10	0	10

Course Level Learning Objectives:

Remember

1. What are the various aspects of licenses?
2. What are the advantages of dual Licensing?
3. What is meant by version control system?
4. What are wikis?
5. List the advantages of using a canned site?
6. Define Dynamic Link Library?

Understand

1. How to use bug tracker for the software?
2. State the uses of change management?
3. Describe the GNU/Linux architecture?
4. Compare between static and shared library?
5. Compare Version and Revision?
6. Summarize the standard set of tools for managing information?

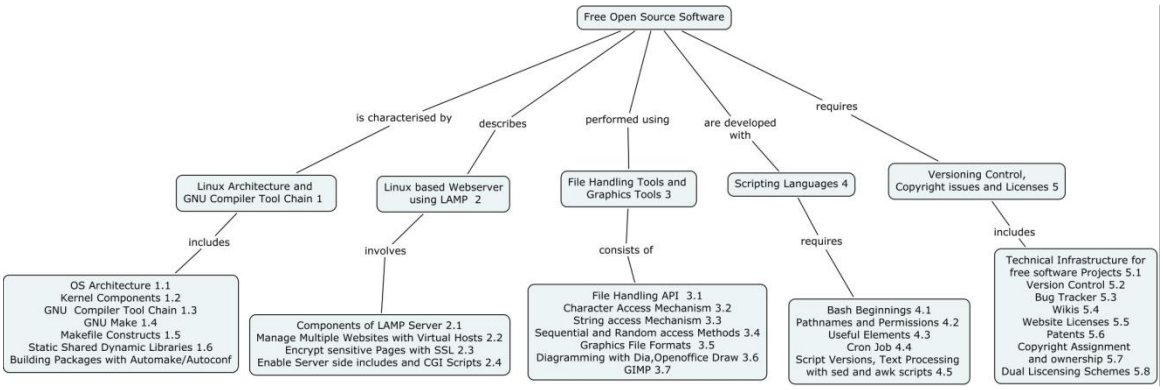
Apply

1. How to use bug tracker for the Mobile Application Software?
2. Using file handling API write a program to read/write a char/variable length string from a file(name)
3. Show how a file can be executed by running the bash command with a hello argument?
4. Check the permissions for a sample hello file?
5. Apply the special built in operators in bash to a file?
6. Show how to take screenshots using GIMP?
7. How to encrypt sensitive pages with SSL?

Create

1. Create a bash script to Find the archive of an given directory?
2. Create an application with directory structure having lib and app directory using autotools? Assume the Directory structure of example project with Autotool files.
3. Design a build script for a Project Directory that has app.c , app.h main.c , lib.c and bar.c ?
4. Create a drawing of the printer using Dia Diagram Window ?

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.

Textbooks:

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	2
	Building packages with Automake/Autoconf	1
2	**Deployment Tools**	
2.1	Components of a LAMP Server,	2
2.2	Manage Multiple Websites with Virtual Hosts,	2
2.3	Encrypt Sensitive Pages with SSL,	2
2.4	Enable Server-side Includes and CGI Scripts.	2
3	**File Handling Tools and Graphics Tools**	
3.1	File Handling-API-Character access mechanisms	2
3.2	String access mechanisms	2
3.3	Sequential and Random access methods	2
3.4	Graphics File Formats,	1
3.5	Diagramming with Dia, Open Office Draw	2
3.6	GIMP	2

S.No.	Topic	No. of Lectures
4	**Text Processing Tools**	
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	**45**

Course Designers:

1. K. Sundarakantham kskcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CCN	3	0	--	3

CCN Storage Infrastructure Management 3:0

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.

Programme Outcomes addressed

a. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

c. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

At the end of the course, the student will be able to

1. Explain the components and functions of information storage systems.
2. Design the storage system for the given specification on interaction with storage user.
3. Investigate the common threats in Storage security domains.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
1	Remember	20	20	20	20
2	Understand	60	40	40	40
3	Apply	20	40	40	40
4	Analyze	0	0	0	0
5	Evaluate	0	0	0	0
6	Create	0	0	0	0

Course Level Learning Objectives

Remember

1. Mention the categories of data.
2. What do you mean by downtime?

3. List the demerits of centralized data storage.
4. Describe the key requirements of storage systems with their functionalities
5. Define Platter.
6. List the challenges of NAS.
7. Define fixed content.
8. Define: Strip and Stripe.
9. Differentiate between Single-mode and Multi-mode fiber.
10. Mention the need for Storage Security.
11. List the benefits of NAS.
12. Mention the services offered by Cloud.

Understand:

1. Describe how you can control Application access, User access and Host access.
2. Explain how CAS stores and retrieves data objects.
3. Mention the benefits of CAS.
4. Describe the evolution of date storage systems.
5. Describe the services offered by Cloud Computing.
6. Describe the functionalities of FC Switch and Hub with necessary diagrams.
7. Describe how you can control Management access and protect Administrative access
8. Discuss the benefits and forms of Virtualization.
9. Compare and contrast NFS and CIFS.
10. Differentiate between Internal and External DAS management
11. Compare and contrast between Integrated and Gateway NAS
12. Explain the IP SAN protocol.
13. Explain how remote replication technology can be helpful in disaster recovery.
14. Describe the Characteristics of Cloud Computing.

Apply

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. Design 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.
4. 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.
5. 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
    b. Compute the preceding parameter if the service time is halved.
6. 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.
7. 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.
8. A manufacturing corporation uses tape as their primary backup storage media throughout the entire organization.

    Current Situation/Issue:

    Full backups are run every Sunday. Incremental backups are run from Monday through Saturday. There are many backup servers in the environment, backing up different groups of servers. Their e-mail and database applications have to be

shut down during the backup process. The main concerns facing the corporation are:

The company would like to:

1) Deploy an easy-to-manage backup environment.

2) Reduce the amount of time the email and database applications need to be shutdown.

3) Reduce the number of tapes required to fully recover a server in case of failure. Construct the topology IP SAN

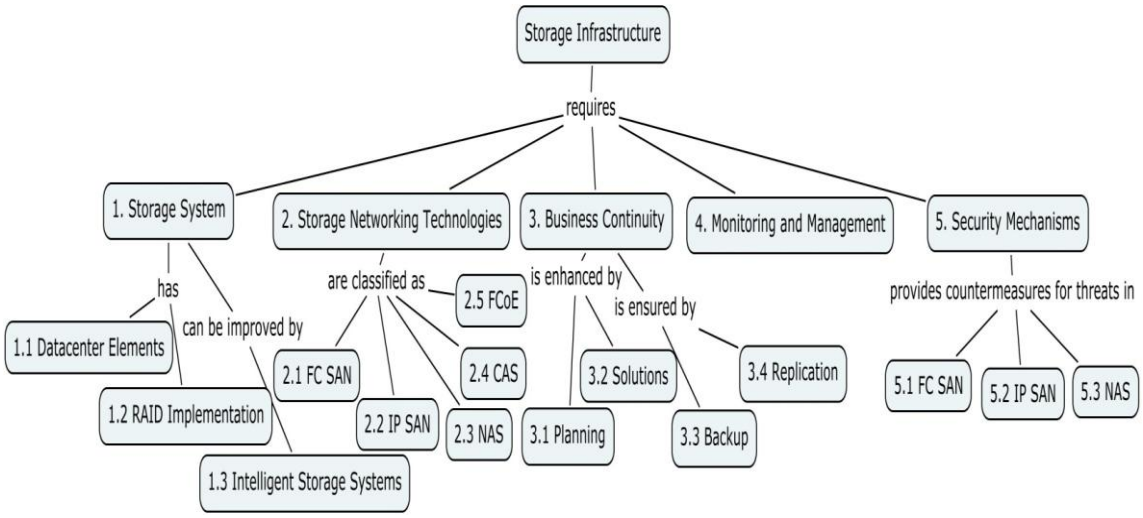
9. A Manufacturing Corporation maintains the storage of their mission critical applications on high-end Storage Arrays on RAID 1 volumes. The corporation has two data centers which are 50 miles apart.

Current Situation/Issue:

The corporation's mission critical Database application takes up 1 TB of storage on a high end Storage Array. In the past year, top management has become extremely concerned because they do not have DR plans which will allow for zero RPO recovery if there is a site failure. The primary DR Site is the 2nd Data Center 50 miles away. The company would like explore remote replication scenarios which will allow for near zero RPO and a minimal RTO. The company is aware of the large costs associated with network bandwidth and would like explore other remote replication technologies in addition to the zero RPO solution.

10. A performance problem has been reported on a database. Monitoring confirms that at 12am, a problem surfaced and access to the database is severely affected until 3pm everyday. This timeslot is critical for business operations and an investigation has been launched. A reporting process that starts at 12pm contends for database resources and constrains the environment. Establish monitoring and management procedures, tools and alerts to ensure accessibility, capacity, performance and security in this environment.

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:

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	3

No.	Topic	No. of Lectures
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	4
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	3
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	3
	Total No. of hours	40

COURSE DESIGNER:

1. Dr.G.S.R. Emil Selvan emil@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1A	1	0	-	1

C1A Rich Internet Applications (RIA) 1:0

Preamble: This course is a one credit course offered for the students of the computer science and engineering. Students gain understanding of how the Internet application works and develop programming skills for developing Rich Internet Applications. This course will establish a professional, client-based and server based attitude towards web-design.

Program Outcomes addressed:

a. An ability to apply knowledge of engineering, information technology, mathematics and science.

b. An ability to design and conduct experiments, as well as to analyze and interpret data.

d.An ability to identify, formulate and solve engineering problems.

Competencies

1. Develop client side programming for web application development.
2. Develop server side programming for web application development
3. Understand how Rich Internet application works using Flash and Flex.
4. Develop programming skills in Rich Internet Applications like AJAX, Flash,Flex.

Assessment Pattern

	Bloom's Category	Test/End-semester examination
	Remember	30
2	Understand	30
3	Apply	40
4	Analyze	0
5	Evaluate	0
6	Create	0

Course Level Learning Objectives:

Remember

1. Flex applications can connect to servers using:
   a. HTTPService
   b. WebService
   c. RemoteObject
d. All of the above

2. The creationComplete event in Flex is:
a. A user event
b. A system event
c. Can be either a user or a system event depending on the component on which it is invoked
d. Neither user nor system event

3. A package for an ActionScript class is:
a. The chain of directory names from source to that class separated by periods.
b. The wrapper class created to access this class
c. The same as every other ActionScript class: "mx.core"
d. None of the above

4. The _____ property of the DateChooser class contains the date clicked on by the user.
a. dateClicked
b.selectedDate
c. date
d. None of the above

5. To navigate in a ViewStack (id="vStack") using a TabBar (id="tBar"):
a. Make the TabBar a child of the ViewStack
b. Make the TabBar the first child of the ViewStack
c. Set the source of the TabBar to "vStack"
d. Set the dataProvider of the TabBar to "vStack"

6. In Flex 4, which of the following properties is used to specify which state the component will be in:
a. addChild
b. includeIn
c. visible
d. None of the above

Understand

1. In which of the following views of Flash Builder 4 would you find application syntax errors?
a. Package explorer
b. Problems View
c. Outline View
d. None of these

2. Which of the following is a valid way to set the maximum decimal digits that a number will accept in the number validator?
a. precision=”0.2”
b. precision=”2”
c. precision=”0.00”
d. All of the above

3. Through which of the following is the new Text Layout Framework (a set of library classes built on top of the Flash text Engine) made available?
a. framework_textLayout.swc
b. textLayout.swc
c. Both of the above
d. None of the above

4. You can also specify styles for specific classes in Flex like for the Button class, you would do it by:
a. @styleName{
color:#686868; src:Button
}
b. s|Button{
color:#686868
}
c. .Button{
color:#686868
}
d. None of the above

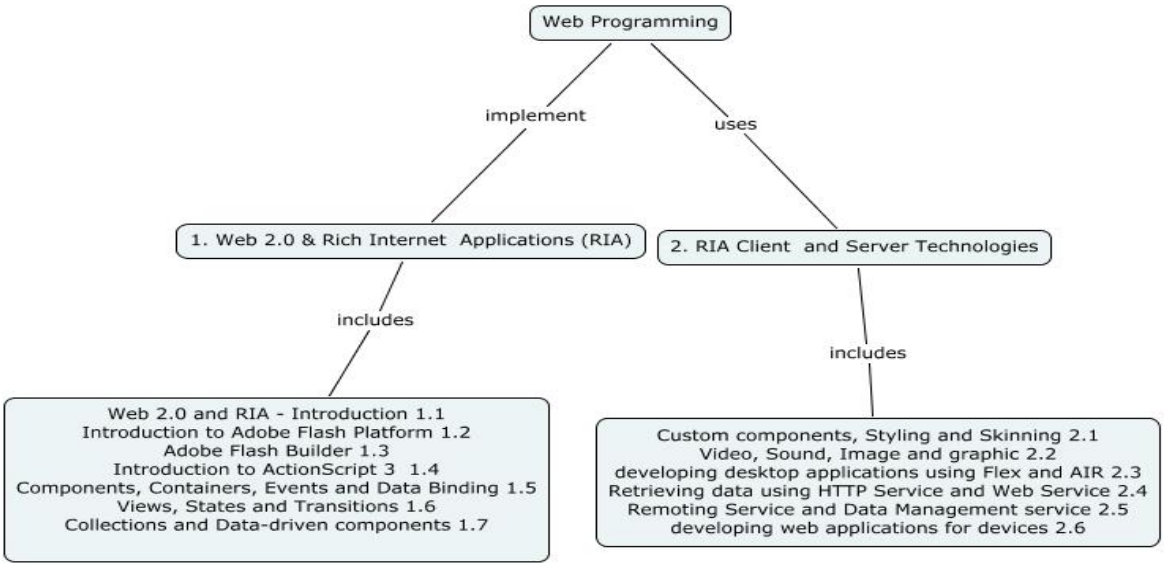
Apply

1. Constructa Photo album application for the following features.
   a. Application should display all images in the album as thumbnails.
   b. When user selects an image, full view of the image should be displayed.
   c. An image will also be associated with a title and some description (200 characters). This data should also be displayed for selected image.

2. Constructa media player application for the following features.
   a. Application should display list of all videos.
   b. When user selects a video from the list, more details of the video should be displayed and user should be able to play the selected video.
   c. User should be able to Play, Pause and Stop the video selected.
   d. Display the time left and total time of video being played.

3. Implementa Flight ticket booking application for the following features.
   a. Allow users to search for a flight options.
   b. Display search results as shown in the mockup image.
   c. Give option to book a flight.
   d. Collect user information when user wants to book ticket.

Concept Map



Course Contents and Lecture Schedule

No	Topic	No of Lectures
1.	**Web 2.0 and Adobe Flash**	
1.1	Web 2.0 and RIA - Introduction	1
1.2	Introduction to Adobe Flash Platform	1
1.3	Getting started with Adobe Flex and Adobe Flash Builder	1
1.4	Introduction to ActionScript 3	1
1.5	Components, Containers, Events and Data Binding	1
1.6	Views, States and Transitions	1
1.7	Collections and Data-driven components	1
2.	**Adobe Flex:**	
2.1	Custom components, Styling and Skinning	1
2.2	Video, Sound, Image and graphic	2
2.3	developing desktop applications using Flex and AIR	1
2.4	Retrieving data using HTTP Service and Web Service	1

2.5	Introduction to Remoting Service and Data Management service	1
2.6	developing web applications for devices	1
	Total	**14**

Syllabus

Web 2.0 and Adobe Flash : Web 2.0 and RIA – Introduction, Introduction to Adobe Flash Platform, Getting started with Adobe Flex and Adobe Flash Builder Introduction to ActionScript 3,Components, Containers, Events and Data Binding, Views, States and Transitions,Collections and Data-driven components.

Adobe Flex: Custom components, Styling and Skinning, Video, Sound, Image and graphic, developing desktop applications using Flex and AIR, Retrieving data using HTTP Service and Web Service, Introduction to Remoting Service and Data Management service, developing web applications for devices.

Text Books:

1. Paul J.Deitel and Harvey M.Deitel, "AJAX, Rich Internet Applications, and Web Development for Programmers", Pearson Education, First Edition, 2009.
2. James Talbot, "Total Training for Adobe Flex 3: Rich Internet &AIR applications", Adobe Incorporations.
3. Jeanette Stallons , Andrew Shorten and Vince, "Getting Started with Flex 4", Adobe Incorporations, 2010.

Course Designers:

1. S. Sridevisridevi@tce.edu
2. R. Suganyarsuganya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1B	1	0	-	1

C1B Advanced Operating Systems 1:0

Preamble: The objective of this course is to help the students learn engineering aspects of developing and sustaining Operating Systems (OS). This Advanced OS course aims to help students develop a deeper understanding of the concepts in OS design they already know. This course teaches a few discrete fundamentals of engineering an OS and associated design principles through practice and some programming.

Prerequisite:

- C44: Operating Systems: Principles and Design
- C34: Operating Systems: Use & Configuration

Program Outcomes addressed

a. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, and mathematics and science problems.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

c. Graduate will demonstrate an ability to engage in life-long learning.

Competencies

The Student will be able to

1. Describe the anatomy of Linux kernel.
2. Apply the OS functionalities to provide Parallel and Distributed computing.
3. Explain the Enterprise Systems like Virtualization, DataCenter convergence.
4. Explain the different types of OS for Mobile/Hand-held Systems.
5. Illustrate the Embedded Systems architecture.
6. Illustrate the Real-time Systems architecture.

Assessment Pattern

	Bloom's Category	Terminal Examination
1	Remember	0
2	Understand	20
3	Apply	40
4	Analyze	30

5	Evaluate	0
6	Create	10

Course Level Learning Objectives:

Understand

1. Understand the organization of Linux Kernel 2.6 source code. Learn building the kernel from source and learn to create and install patch in Linux.
2. Observe Linux behavior by writing a program to use the /proc mechanism to inspect various kernel values such as processor type, kernel version, time since booting, memory configured, memory currently available and many more.
3. Understand the steps taken place during the booting process of the kernel.

Apply

1. Write a simple program that implements a binary tree data structure in memory. This program should allow insert, delete and update of nodes. Each node is a record that has a unique id of type unsigned long (e.g. student roll number) and associated data of size 256 bytes. Demonstrate the working of this program for a maximum data size of 16 records.
2. Extend the program written above to make the binary tree persistent. This means that the records are stored in the disk. Study the scalability of this program and improve.
3. Write a program using multi-threading to print 1, 2, 3, 4, etc. each number output by different threads. Initially implement the program to print odd numbers from one thread and even numbers from the other thread. Extend the program to print from several threads but in sequence.

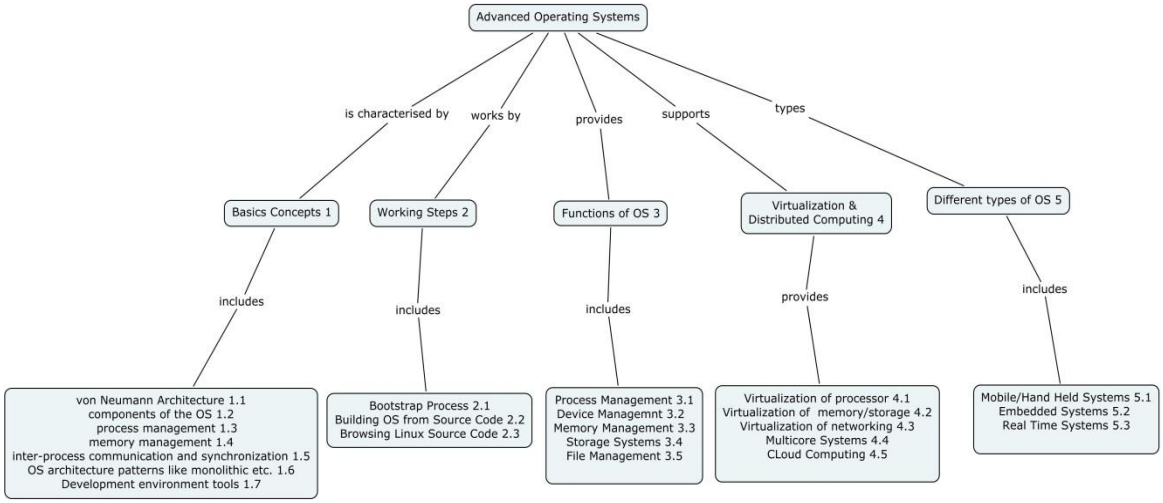
Analyze

1. Analyze Kernel Virtual Machine (KVM) and implement virtual machines in Linux.
2. Study the source code for a simple device driver in Linux.
3. Download and learn Qemu PC emulator
4. Download the Linux Kernel source code. Set up the source code using cscope or equivalent/better source code browsing tool.
5. Compare the different ways to design the Operating System?

Create

1. Design and implement a program that uses the Linux timers to measure processor usage of a process and prepare performance report of a process.
2. Develop a simple shell command interpreter program. Implement piping operation.
3. Develop a simple kernel module, to dynamically add functionality to the Linux kernel.
4. Develop/modify a File System module in Linux.
5. Implement a simple file manager program to lists the directories and files in a file system; facilitates renaming files and directories; deletes files and directories. The user interface of this program is to be implemented as a separate program. The file manager and the user interface program to communicate using RPC. Execute the user interface from a different computer and prove working of the solution.

Concept Map:



Syllabus

Review of Basic Concepts: Present a programmer's view of the OS. Review the basic concepts of OS, von Neumann Architecture, components of the OS, process management, memory management, inter-process communication and synchronization; OS architecture patterns like monolithic, layered, micro-kernel etc., Development environment tools such as cscope,gdb, GNU compiler tools, patch.**Working with an operating system:** Bootstrap process, Building an OS from

source code, Browsing Linux source code.**Process Management:** Notion of process, Multi-programming, Time sharing, Concept of timer,ProcessStates, Scheduling, Context Switching.**Device Management:** Understand the essential elements of device management principles, I/O principles with special focus on Disk I/O devices.**Memory Management and Storage Systems:** Memory Management, Storage, data and information systems, Storage Infrastructure and Management, Information Life Cycle Management, System architectures in storage systems, storage gateways, data protection systems. **File Management:** Understanding of file systems and management, Study of a few modern file systems and their properties, network file systems, file system properties and features of modern enterprise-class file systems. **Virtualization:** Virtualization principles and trends, Kernel Virtual Machine (KVM), Virtualization of processor, memory/storage and networking, Requirements of enterprise data center, high availability, convergence of data center infrastructure.**Distributed Computing:** Architecture principles and OS design considerations for distributed computing systems, Flynn's classification, Enabling mechanisms, Distributed computing in practice: multi-core systems, An overview of the scale-up and scale-out architectures and utility computing trends, load-balancing, scalability, cluster computing, high availability and fault tolerance, Architecture issues in shared infrastructure or cloud computing.**Operating Systems for Mobile/Hand-held Systems:** Technology drivers and special requirements of mobile/hand-held systems such as smart-phones, study of a few OS architectures for mobile/hand-held systems such as Symbian™, Android™, MeeGo™, WebOS™. **Operating Systems principles for Embedded Systems:** Basics of embedded systems, Concepts and architectures from OS perspective, hardware-software partitioning of concerns.**Operating Systems principles for Real-time:** Special requirements of real-time systems, Data Acquisition and Control Systems, Sensor Networks

Text Book:

1. Pramod Chandra P. Bhatt, "An Introduction to Operating Systems: Concepts and Practice," 3rd Edition, PHI Learning, ISBN 978-81-203-4138-8.

References:

1. Gary Nutt, "Operating Systems," Third Edition, Pearson Education Low Price Edition, 2004, ISBN 81-297-0614-8.

2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Principles", 7th Edition, Wiley India, 2006, ISBN 81-265-0962-7.
3. William Stallings, "Operating Systems: Internals and Design Perspectives," Fifth Edition, Prentice-Hall of India Eastern Economy Edition,2005,ISBN 81-203-2796-9.
4. Andrew S. Tanenbaum, Albert S. Woodhull, "Operating Systems: Design and Implementation, the MINIX Book" Third Edition, Person Prentice-Hall Edition, 2009, ISBN 978-81-317-0514-8.
5. Robert Love, "Linux Kernel Development," 3rd Edition, Person Education, 2010, ISBN 978-81-317-5818-2.
6. Daniel P. Bovet & Marco Cesati, "Understanding the Linux Kernel", 3rd Edition, O'Reilly, Shroff Publishers & Distributors, 2008, ISBN 81-8404-083-0.
7. Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman, "Linux Device Drivers," Third Edition, URL: http://lwn.net/Kernel/LDD3/.
8. Pradeep K. Sinha, "Distributed Operating Systems: Concepts and Design," PHI Learning, 2007, ISBN 978-81-203-1380-4.
9. MukeshSinghal, Niranjan G. Shivaratri, "Advanced Concepts in Operating Systems," Tata McGraw Hill, 2001, ISBN 0-07-047268-8.
10. George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems: Concepts and Design," 4th Edition, Person Education, 2009, ISBN 978-81-317-1840-7.

Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Review of Basic Concepts**	
1.1	Present a programmer's view of the OS. Review the basic concepts of OS, von Neumann Architecture, components of the OS, process management, memory management, inter-process communication and synchronization.	1
1.2	OS architecture patterns like monolithic, layered, micro-kernel etc., Development environment tools such as cscope,gdb, GNU compiler tools, patch.	1
2	**Working with an operating system**	
2.1	Bootstrap process,Building an OS from source code,	1

	Browsing Linux source code.	
3	**Process Management**	
3.1	Notion of process, Multi-programming, Time sharing, Concept of timer, Process States, Scheduling, Context Switching.	1
4	**Device Management**	
4.1	Understand the essential elements of device management principles, I/O principles with special focus on Disk I/O devices.	1
5	**Memory Management and Storage Systems**	
5.1	Memory Management, Storage, data and information systems, Storage Infrastructure and Management, Information Life Cycle Management, System architectures in storage systems, storage gateways, data protection systems.	2
6	**File Management**	
6.1	Understanding of file systems and management, Study of a few modern file systems and their properties, network file systems, file system properties and features of modern enterprise-class file systems.	1
7	**Virtualization**	
7.1	Virtualization principles and trends, Kernel Virtual Machine (KVM), Virtualization of processor, memory/storage and networking, Requirements of enterprise data center, high availability, and convergence of data center infrastructure.	1
8	**Distributed Computing**	
8.1	Architecture principles and OS design considerations for distributed computing systems, Flynn's classification, Enabling mechanisms, Distributed computing in practice: multi-core systems	1
8.2	An overview of the scale-up and scale-out architectures and utility computing trends, load-balancing, scalability, cluster computing, high availability and fault tolerance, Architecture issues in shared infrastructure or cloud computing.	1
9	**Operating Systems for Mobile/Hand-held**	

	Systems	
9.1	Technology drivers and special requirements of mobile/hand-held systems such as smart-phones, study of a few OS architectures for mobile/hand-held systems such as Symbian™, Android™, MeeGo™, WebOS™.	2
10	**Operating Systems principles for Embedded Systems**	
10.1	Basics of embedded systems, Concepts and architectures from OS perspective, hardware and software partitioning of concerns.	1
11	**Operating Systems principles for Real-time**	
11.1	Special requirements of real-time systems, Data Acquisition and Control Systems, Sensor Networks	1
	Total No of Hours	15

Course Designer:

1. Sundara Nagarajan - s.nagarajan@computer.org
2. G.MadhuPriya – gmadhupriya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1C	1	0	-	1

C1C Mobile Application Development 1:0

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:

- C22 – Computers and Programming
- C42 – Computer Networks: use and configuration

Program Outcomes addressed:

c. An ability to design a system or component, or process to meet stated specifications.

e. An ability to use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

h. An ability to engage in life-long learning.

Competencies:

Student will be able to

1. Describe the major mobile device platforms and their capabilities.
2. Apply the mobile platform SDK and develop applications.
3. Explain the patterns for integration with enterprise systems and cloud services.
4. Illustrate the Mobile Application architecture.
5. Apply techniques to leverage the native capabilities of the device.

Assessment Pattern:

	Bloom's Category	End-semester examination
1	Remember	0
2	Understand	50
3	Apply	30
4	Analyze	0
5	Evaluate	0
6	Create	20

Course Level Learning Objectives:

Understand

1. Describe the Apple (iOS) and Android Ecosystem.
2. Summarize the various technology options available for development like native development, cross platform development platforms, rich web.
3. Explain the capabilities of platform SDK and rich web technologies.
4. Discuss the enterprise integration technologies.
5. Discuss the various Integrated Development Environments (IDEs) available for mobile application development.
6. Report the device capabilities of Android and iOS based smart phones and Tablets.

Apply

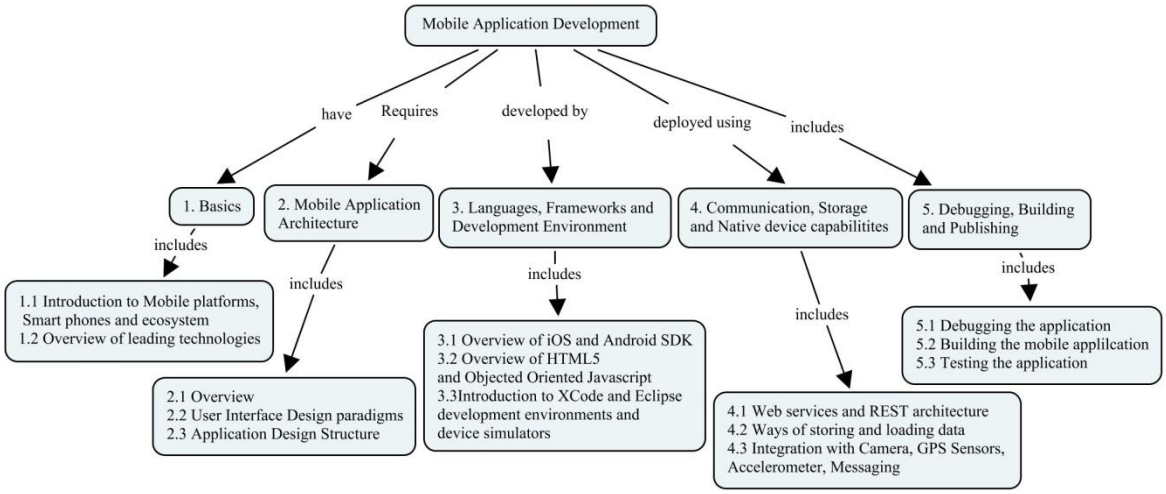
1. Write a simple iOS notes application to take notes based on the text entered by the user and store notes in a database.
2. Write the same notes application using Android SDK and HTML5.
3. Practice XCode IDE on Apple Mac Desktop.
4. Practice Eclipse IDE on Windows Desktop.
5. Use the online guides from Apple and Google on iOS and Android SDK.
6. Illustrate the different ways to design a mobile application.

Create

1. Develop a REST service to receive data from the notes mobile application and save it on a server database.
2. Enhance the notes mobile application to communicate with the server over REST protocol and save the notes on the server.

3. Use smart phone's camera to take a photo and attach a note that shows the location of photo. [HINT: Use GPS available on the smart phone].

Concept map:



Course contents and Lecture Schedule:

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	2
3.2	Overview of HTML5 and Objected Oriented	1

	Javascript	
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	2
5.2	Building the mobile app and publishing to the market	1
5.3	Testing the mobile app	1
	Total No of Hours	16

Syllabus:

Basic Concepts: Introduction to mobile platforms, smart phones and ecosystem Overview of leading technologies available for mobile 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", 1stEditon, 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 Designers:

1. Gangadhar Neeli Gangadhar.Neeli@honeywell.com
2. SubramanianKrishnansubramanian.krishnan@honeywell.com
3. M. Sugunamscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1D	1	0	-	1

C1D Basics of Application Security 1:0

Preamble: Application Security refers to the technique which makes use of software, hardware and procedural methods to prevent security flaws inherent in the application thereby prescribe solutions to prevent/protect the applications from the external or internal threats. 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 unauthorized means.

Prerequisite

- C55: Web Programming
- C36:Oject Oriented Programming

Program Outcomes Addressed

c. an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

k. an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

f. an understanding of ethical and professional responsibilities.

Competencies

At the end of the course the Student will be able to

1. List the application threats
2. Understand the concepts of Information and Application security.
3. Apply the secure coding principles while developing real time applications.
4. Analyze different types of attacks on web applications.
5. Rate the security of an application
6. Propose a security solution.

Assessment Pattern

	Bloom's Category	End Semester Examination
1	Remember	10
2	Understand	20
3	Apply	30
4	Analyze	20
5	Evaluate	10
6	Create	10

Course Level Learning Objectives:

Remember

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?

Understand

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?

Apply

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 vulnerable demo application
4. Construct a secure database for a web application.
5. Contrast the SDLC framework and secure SDLC framework.

Analyze

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.

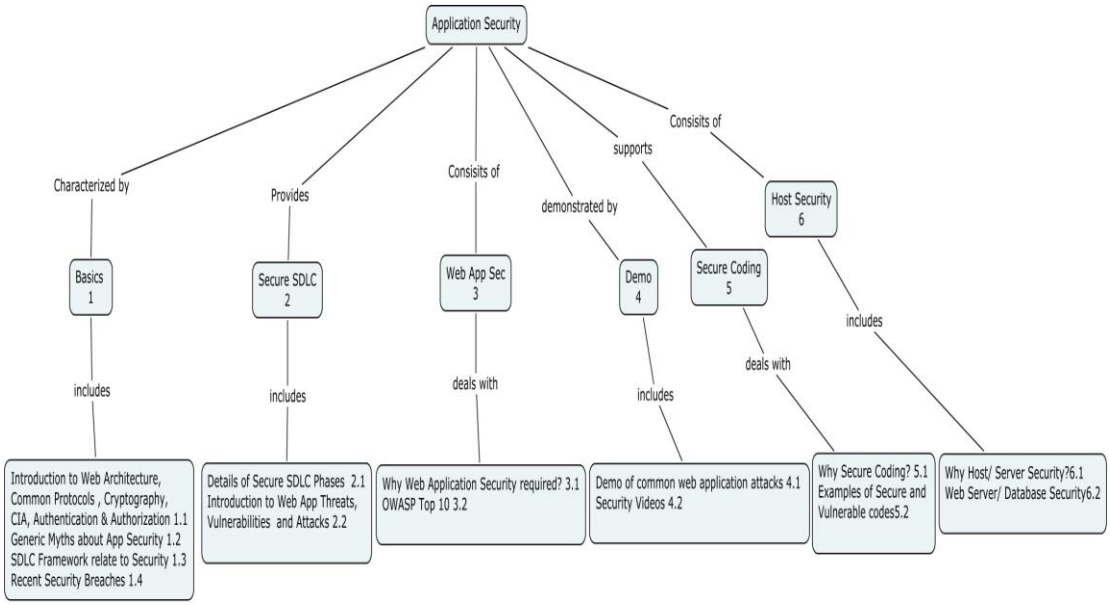
Evaluate

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 in the following code.
   SELECT booktitle **FROM** booklist **WHERE** bookId = '00k14cd' **AND** '1'='1'

Create

1. Incorporating the secure SDLC process, design and implement a simple program that collects employee information from the end users and stores it on to a database and retrieves it back when requested by the end user.
2. Create/Design attack vectors for the common web application vulnerabilities.
3. Develop Configuration level checklists for a windows based operating system's infrastructure review.
4. Create functions or routines that can sanitize the user inputs and encode them.
5. Incorporating the secure SDLC process, design a simple program for Railway Reservation System.

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

References:

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,ISBN-13:9780132126953
4. OWASP Security Testing Guide (https://www.owasp.org/index.php/File:OWASP_Testing_Guide_v2_pdf.zip)

5. OWASP: http://www.owasp.org
6. WASC: http://www.webappsec.org/
7. 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	15

Course Designers:

1. Mr. Satheesh PRV - Satheesh.Kumar.P.R.Veerapadran@honeywell.com
2. Mr. Ravikanth Dangeti – Ravinkanth.Dangeti@honeywell.com
3. Mr. Akash Shrivastava - Akash.Shrivastava@Honeywell.com
4. Mr. Ramesh Sitaraman - Ramesh.Sitaraman@Honeywell.com
5. Mr. Murali Krishnan - Murali.KrishnanS@Honeywell.com
6. Mrs. M.Vijayalakshmi mviji@tce.edu

Sub Code	Lecture	Tutorial	Practical	Credits
C1E	1	0	-	1

C1E Cyber Security 1:0

Preamble:

The objective of this course is to update the knowledge and skills in cyber security to empower the students in understanding the networking, security concepts, threats and vulnerabilities and thereby making them implement the countermeasures to safe guard the resources.

Prerequisite

- C42:Computer Networks: Use and Configuration

Programme Outcomes addressed

e. An ability to identify, formulate, and solve engineering problems

k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Competencies

At the end of the course the student will be able to

1. Describe the information security governance.
2. Summarize the external and internal threats to an organization.
3. Explain the protocols for security services.
4. Diagnose the network security threats and countermeasures.
5. Examine the vulnerabilities and security solution.

Assessment Pattern

	Bloom's Category	End-semester examination
1	Remember	20
2	Understand	20
3	Apply	20
4	Analyze	40
5	Evaluate	0
6	Create	0

Course Level Learning Objectives

Remember

1. Recall the types of firewall.
2. What is port scanning attack?
3. What is the significance of network mapping?
4. What does it mean to lock down a computer?
5. What is brute force attack?

Understand

1. How to detect brute force attack?
2. Illustrate how can Email Spoofing be useful?
3. How encryption ensures message integrity?
4. Indicate how the tribe flood network attack affects the victim?
5. Report the security features provided in wireless networks?

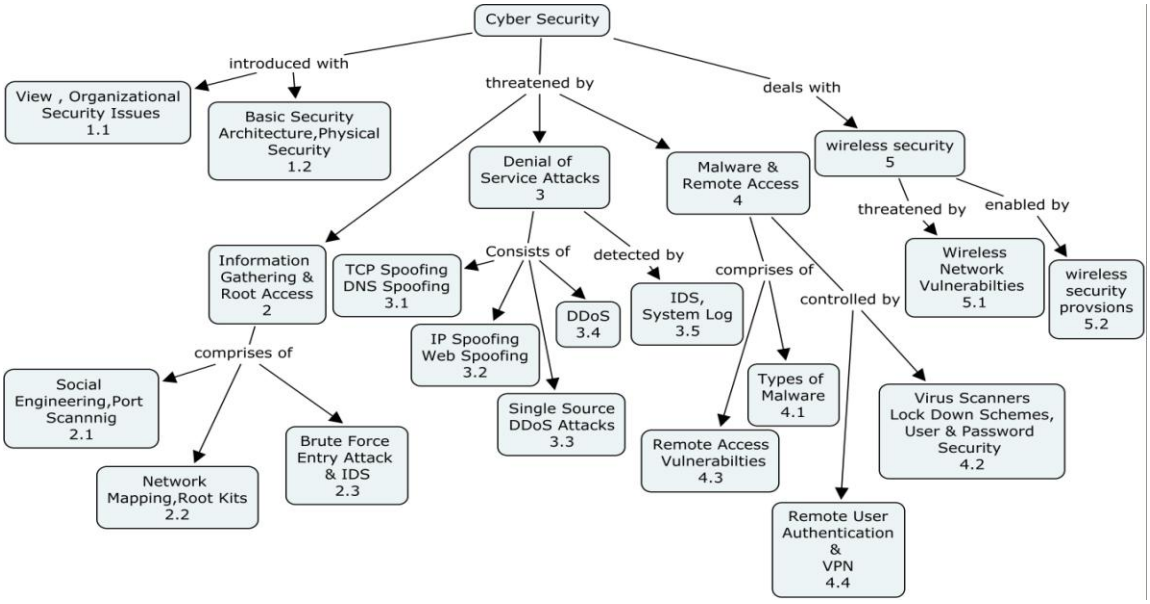
Apply

1. Illustrate the different techniques that can be used to protect the network against information gathering.
2. Calculate the maximum amount of time that it would take for a computer that can try 1 million RC2 keys every second to do a brute-force attack on a Microsoft Office document sealed with a cryptographic transformation that uses a 40-bit shared secret.
3. Compute the SHA-1 of the string "Thiagarajar College of Engineering."
4. Criticize firewall vulnerabilities.
5. Illustrate buffer overflow attack with an example.

Analyze

1. Compare Host based Intrusion detection system and Network based Intrusion Detection System.
2. Why TCP Spoofing is difficult to detect?
3. Differentiate the types of Malware based on the propagation methods.
4. What are the advantages of RSA secureID?
5. Analyze the remote access vulnerabilities.
6. Analyze the security vulnerabilities of RADIUS.
7. Analyze the different remote user authentication protocols.
8. Analyze the vulnerabilities of Wireless network.

Concept Map



Syllabus

INTRODUCTION - Views of Network Security, Organizational Security Process, Preparing Security Policy, Security Audits, Basic Security Architecture – Secure Network Layouts, Firewalls, File Permissions, Physical Security – Protecting System Console, Manage System Failure. **INFORMATION GATHERING AND ROOT ACCESS -** Social Engineering, Port Scanning, Network Mapping, Root Kits, Brute Force Entry Attacks and Intrusion Detection, Buffer Overflow Attacks. **DENIAL OF SERVICE ATTACKS -** TCP Spoofing, DNS Spoofing, IP Spoofing, Web Spoofing, Single Source DoS Attacks, Distributed DoS Attacks, Using IDS to detect a DoS Attack, Using System Logs to Detect a DoS Attack. **MALWARE AND REMOTE ACCESS -** Types of Malware, Virus Scanners, Dealing with Removable Media, Lockdown Schemes, User and Password Security – Password Policy , Password Audit, Remote Access Vulnerabilities, VPNs, Remote User Authentication. **WIRELESS SECURITY -** Wireless Standards, Wireless Network Vulnerabilities, Wireless Security Provisions.

References:

1. Jan L.Harrington,"Network Security – A Practical Approach", Morgan Kaufmann Publishers –An Imprint of Elsevier, 2005, ISBN 13:978-0-12-311633-8.

2. William Stallings, "Cryptography and Network Security – Principles and Practice", Pearson Education Asia, Fourth Edition,2005,ISBN -13:9780131873162.
3. Edward Amoroso, "Cyber Security", Silicon Press, 2006, ISBN 0-929306-38-4.

Course plan and Lecture Schedule

No	Topic	No of Lectures
1.	**INTRODUCTION (2)**	
1.1	Views of Network Security, Organizational Security Process, Preparing Security Policy, Security Audits	1
1.2	Basic Security Architecture – Secure Network Layouts, Firewalls, File Permissions, Physical Security – Protecting System Console, Manage System Failure	1
2	**INFORMATION GATHERING AND ROOT ACCESS (3)**	
2.1	Social Engineering, Port Scanning	1
2.2	Network Mapping , Root Kits	1
2.3	Brute Force Entry Attacks , Buffer Overflow Attacks and Intrusion Detection	1
3.	**DENIAL OF SERVICE ATTACKS (5)**	
3.1	TCP Spoofing, DNS Spoofing	1
3.2	IP Spoofing, Web Spoofing	1
3.3	Single Source DoS Attacks	1
3.4	Distributed DoS Attacks	1
3.5	Using an IDS to detect a DoS Attack, Using System Logs to Detect a DoS Attack	1
4	**MALWARE AND REMOTE ACCESS (4)**	
4.1	Types of Malware	1
4.2	Virus Scanners, Dealing with Removable Media, Lockdown Schemes, User and Password Security – Password Policy , Password Audit	1

4.3	Remote Access Vulnerabilities	1
4.4	VPNs, Remote User Authentication	1
5	**WIRELESS SECURITY (2)**	
5.1	Wireless Standards, Wireless Network Vulnerabilities	1
5.2	Wireless Security Provisions	1
	TOTAL	16

Course Designers:

1. Dr. S. V. Raghavan, IITM, Chennai
2. Mrs. M.Vijayalakshmi mviji@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1F	1	0	-	1

C1F Green Data Center 1:0

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 practice and some Labs

Prerequisite:

- C44: Operating System: Principles and Design
- C53:Multicore Architecture

Program Outcomes addressed

This course aims at providing an overview of the state-of-the-art directions in the Green Data Center evolution. 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. Leaning objectives for the students are:

- Familiarize with different types of virtualization and understand the important of virtualization for Green Data Center.
- Familiarize with the various metrics used to define and calculate the datacenter energy metrics.
- Describe a typical Data Center (DC), introduce DC problems faced in today's DC's, and explain the need for an Energy efficient Green DC, ways to achieve Green DC goals and Future technologies in DC.
- Familiarize with various tools and technologies used to optimize the cost of datacenter operations
- Students can setup open source virtualization components, Qemu and KVM (Kernel-based virtual machine) in a VT (Virtualization Technology) enabled system
- Install Virtual Machines to understand how hardware assisted virtualization works.

Competencies

The following themes will be covered as part of this course:

- Describe typical Green Data center

- Importance of virtualization for Green Data Center
- Basics of active energy management and cost optimization
- Explain the ways of optimizing the IT Energy Consumption
- Use various tools and technologies used to optimize the cost of datacenter operations
- Evaluate the ways to achieve Green Data Center goals and Future technologies in DC

Successful students of this course would have developed a basic level of competency to contribute how to build and maintain systems while reducing its negative impact on the environment. This competency will enable them to pursue engineering and R&D careers in the computer systems and embedded systems industry.

Assessment Pattern

	Bloom's Category	**Terminal Examination**
1	Remember	0
2	Understand	30
3	Apply	50
4	Analyze	0
5	Evaluate	20
6	Create	0

Course Level Learning Objectives:

Understand:

1. Explain the need for Datacenter.
2. Classify the types of virtualization.
3. Associate the impact of virtualization to Green Data Center.
4. Report the ways to reduce active power in the CPU at run time.
5. Discuss the different ways to reduce the power in memory and I/O.
6. Report the ways of shutting off unused cores.

Apply:

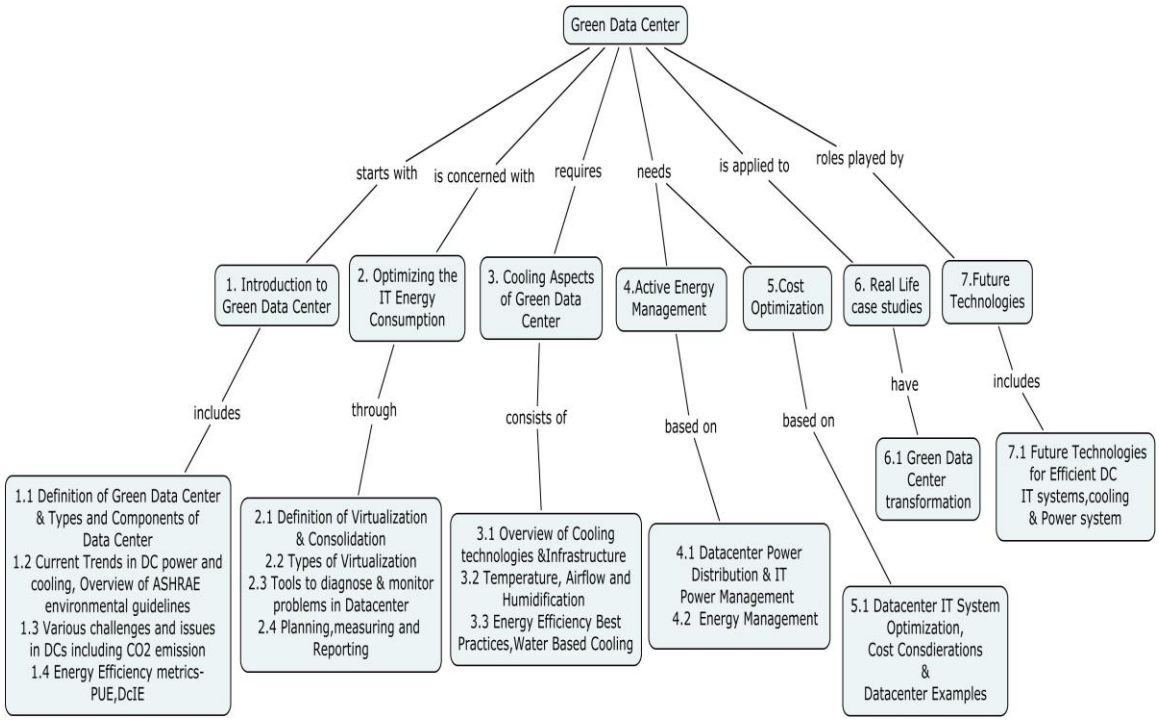
1. Demonstrate various ways of reducing power in memory and IO.
2. Apply ASHRAE environment guidelines to maintain temperature and humidity for the IT systems in a datacenter.

3. Use QEMU, and allowing applications compiled for one architecture to be run on another
4. Use pm-utils, to suspend and hibernate the Computer
5. Practice Power top in Linux Machines to observe the performance of system.
6. Sketch the various components of Data center.
7. Sketch various ways to achieve Green DC goals.

Evaluate:

1. Evaluate the ways to monitor and manage the server/IT system power consumption using management tools.
2. Estimate the active power of the Intel laptop using power top command
3. Measure the CPU utilization for the running and runnable programs in Linux system.
4. Evaluate the performance of KVM Emulator.
5. Predict the status of the UPS using NUT package.
6. Measure the performance of the process in a Linux OS using top command.

Concept Map:



Syllabus

Introduction to Green Datacenter: Typical Definition of DC, need for a datacenter, various components in a datacenter, Power and cooling trends in a datacenter, the history and future of IT power footprint, ASHRAE environmental guidelines to maintain temperature and humidity/dewpoint for the IT systems in a datacenter,

Problems in datacenter and challenges faced by IT and facilities managers in a datacenter, various metrics used to define energy efficiency of a datacenter and ways to calculate and monitor the datacenter energy metrics. **Optimizing the IT Energy Consumption:** Introduction to Green IT, How to reduce active power in the CPU at run time, Shutting off unused cores, Moderating speculation in the machine Power-down states, Inactive cores various power down states and latency impact of waking them up, Reducing power in Memory and IO. Definition of Virtualization, Hardware virtualization, Platform virtualization, Full virtualization, Para virtualization, Hardware assisted virtualization, Operating System level virtualization, Application Virtualization, Network Virtualization, Storage virtualization, Definition of Green IT, Relation between Green IT and Energy Efficient computing, Deployment Optimization and Relevance of Virtualization in Green IT. **Cooling Aspects of Green Datacenter**:Need for maintaining temperature and humidity at specific levels in a datacenter, Airflow distribution in datacenter, Various technologies for cooling IT systems in a datacenter an overview, Cooling best practices an overview and future with liquid cooled systems. **Active Energy Management:** Power distribution systems in a datacenter like UPS system, PDU systems and ways to monitor and manage the power distribution systems, Overview of losses in a power distribution system and its effect in datacenter energy efficiency, Monitoring and managing the server/IT system power consumption using various management tools. **Cost optimization:** Overview of various tools and technologies used to optimize the cost of datacenter operations, Various measurement and monitoring tools used to reduce the datacenter energy consumption and cost. **Real Life Case Studies:** Videos and presentation on Green Datacenter case studies. **Future Technologies:** Future of IT systems and its computing performance, Cooling systems like free cooling, direct water cooling to servers and datacenter infrastructure.

References:

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 content and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction to Green Datacenter(4)**	
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	Various challenges and issues in DCs including CO2 emission	1
1.4	Energy Efficiency metrics – Power usage Effectiveness (PUE), Datacenter Infrastructure Efficiences (DcIE)	1
2	**Optimizing the IT Energy Consumption (4)**	
2.1	Definition of Virtualization and Consolidation	1
2.2	Types of Virtualization	1
2.3	Tools to diagnose & monitor problems in Data	1

	center	
2.4	Planning, measuring and Reporting	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 (2)**	
4.1	Datacenter Power Distribution and IT Power Management	1
4.2	Energy Management	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. Mr. Vidya Shankar & Dr. Sampath Na Parthasarathy
   sambath.narayanan@in.ibm.com
2. Mr. Vaidyanathan & Mr.Dipankar - IBM India Green IT Team
3. Dr.S.Padmavathi – spmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1G	1	-	-	1

C1G: Introduction to Avionics 1:0

(Common to Mechanical, EEE, ECE, CSE and IT Departments)

Preamble:

This course is offered as interdisciplinary course. Avionics refer to the electronic systems used on aircraft, artificial satellites, and spacecraft. Avionic systems include communications, navigation, the display and management of multiple systems, and the hundreds of systems that are fitted to aircraft to perform individual functions. These can be as simple as a searchlight for a police helicopter or as complicated as the tactical system for an airborne early warning platform.

Program outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.
b. Graduates will demonstrate an ability to design a system, components or process as per needs and specifications.
c. Graduates will demonstrate skills to use modern engineering tools, software's and equipment to analyze problems.

Competencies:

At the end of the course the student should be able to

1. Identify the need and various sub systems for representing avionics systems
2. Understand the concepts involved in Avionics system design.
3. Understand the requirements expected for aircraft standard.
4. Understand the essential display components and maintenance procedure related to avionics.

Assessment pattern

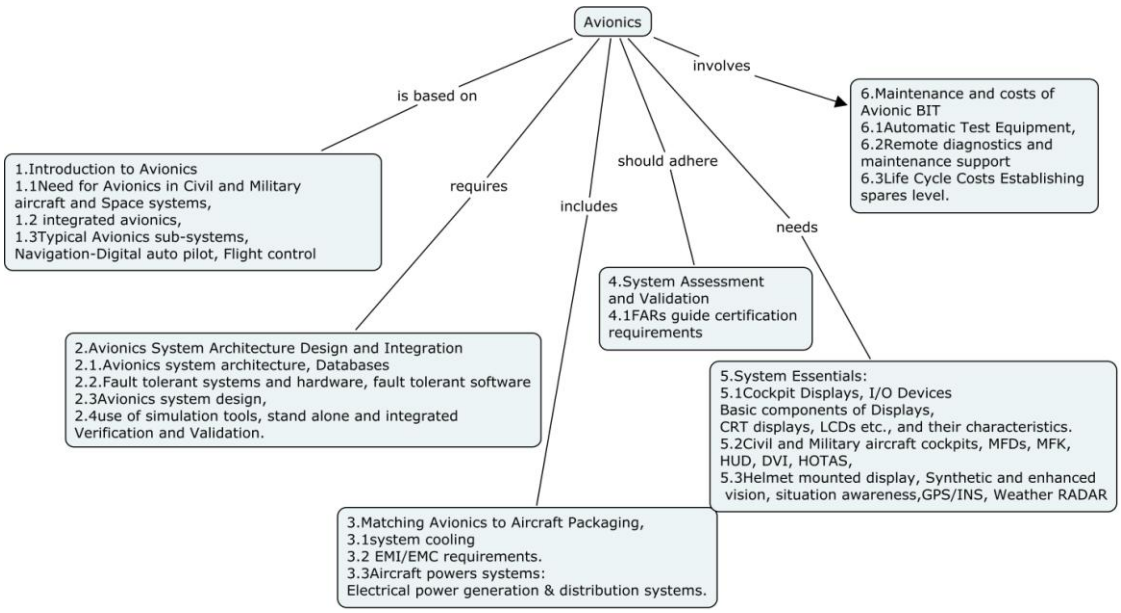
	Bloom's category	**End-semester Examination**
1	Remember	40
2	Understand	60
3	Apply	-
4	Analyze	-

5	Evaluate	-
6	Create	-

Examination pattern

End semester Examination will be conducted in an objective type question pattern.

Concept Map



Syllabus

Introduction to Avionics Need for Avionics in Civil and Military aircraft and Space systems, integrated avionics, Typical Avionics sub-systems, Navigation –Digital auto pilot , Flight control **Avionics System Architecture Design and Integration** Avionics system architecture, Databases, Fault tolerant systems and hardware, fault tolerant software. Avionics system design, use of simulation tools, stand alone and integrated Verification and Validation.**Matching Avionics to Aircraft** Packaging, system cooling, EMI/EMC requirements. Aircraft powers systems: Electrical power generation & distribution systems.**System Assessment and Validation** FARs guide certification requirements **System Essentials: Cockpit Displays, I/O Devices** Basic components of Displays, CRT displays, LCDs etc., and their characteristics. Civil and Military aircraft cockpits, MFDs, MFK, HUD, DVI, HOTAS, Helmet mounted display, Synthetic and enhanced vision, situation awareness,

GPS/INS, Weather RADAR **Maintenance and costs of Avionic** BIT, Automatic Test Equipment, Remote diagnostics and maintenance support-Life Cycle Costs Establishing spares level.

References:

1. Middleton, D.H. 'Avionics Systems', Longman Scientific and Technical, Longman Group UK Ltd., England, 1989.
2. Spitzer, C.R. 'Digital Avionics Systems', Prentice Hall, Englewood Cliffs, N.J., U.S.A.,1987.
3. Collinson R.P.G. 'Introduction to Avionics', Chapman and Hall, 1996
4. Jim Curren, Trend in Advanced Avionics, IOWA State University, 1992.
5. Cary R. Spitzer, The Avionics Handbook, Crc Press, 2000.

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1.	**Introduction to Avionics**	
1.1	Need for Avionics in Civil and Military aircraft and Space systems	1
1.2	Integrated avionics	1
1.3	Typical Avionics sub-systems, Navigation –Digital auto pilot , Flight control	
2.	**Avionics System Architecture Design and Integration**	
2.1	Avionics system architecture, Databases	1
2.2	Fault tolerant systems and hardware, fault tolerant software.	1
2.3	Avionics system design	1
2.4	use of simulation tools, stand alone and integrated Verification and Validation.	1
3.	**Matching Avionics to Aircraft**	
3.1	Packaging, system cooling	1
3.2	EMI/EMC requirements.	1
3.3	Aircraft powers systems: Electrical power generation & distribution systems.	1
4	**System Assessment and Validation**	
4.1	FARs guide certification requirements	2
5.	**System Essentials: Cockpit Displays, I/O Devices**	

5.1	Basic components of Displays, CRT displays, LCDs etc., and their characteristics.	1
5.2	Civil and Military aircraft cockpits, MFDs, MFK, HUD, DVI, HOTAS, Helmet mounted display	1
5.3	Helmet mounted display ,Synthetic and enhanced vision, situation awareness, GPS/INS, Weather RADAR	
6	**Maintenance and costs of Avionic**	
6.1	BIT, Automatic Test Equipment	1
6.2	Remote diagnostics and maintenance support	1
6.3	Life Cycle Costs Establishing spares level.	1
	Total No. of Lectures	16

Course Designers:

1. Krishnaswamy Rajagopal Rajagopal.Krishnaswamy@honeywell.com
2. Saravanan Jayachandran Saravanan.jayachandran@honeywell.com
3. Soundar Arunachalam Soundar.Arunachalam@honeywell.com
4. Hariharan Saptharishi hariharan.saptharishi@honeywell.com
5. Chandrasekaran Babu *Chandrasekaran.Babu*@honeywell.com
6. Ari Narayanan Ari.Narayanan@honeywell.com
7. Dr.K.Hariharan khh@tce.edu
8. C.Senthilkumar cskcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1H	1	0	-	1

C1H - Introduction to IT Service Management 1: 0

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.

Program Outcomes addressed

a. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.
b. Graduates will demonstrate an ability to visualize and work on laboratory and multi-disciplinary tasks.
c. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.
d. Graduate will show the understanding of impact of engineering solutions on the society and also will be aware of contemporary issues.

Competencies

At the end of the course the student should be able to

1. Understand IT department of an organization manages the business needs and expectations from strategy to implementation and daily management of IT as a service.
2. Understand and design the ITIL life cycle for IT solutions
3. Understand the service transition and operation, continuous service improvement requirement for an IT organisation

Assessment Pattern

	Bloom's Category	Terminal Examination
1	Remember	10
2	Understand	30
3	Apply	60

4	Analyze	0
5	Evaluate	0
6	Create	0

Course Level Learning Objectives:

Remember:

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.

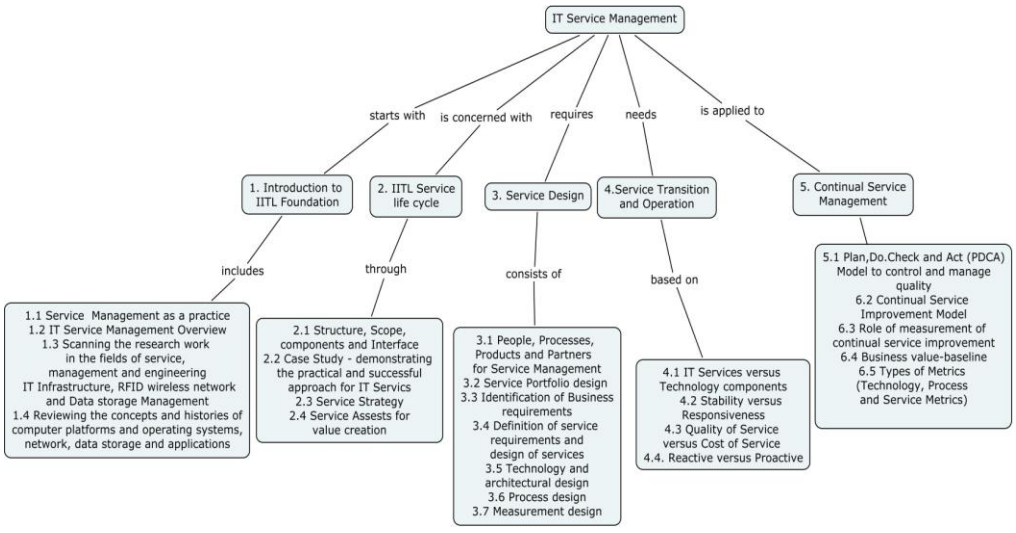
Understand:

1. Outline the overview of IT service management.
2. Explain about various metrics used in service measurement.
3. Describe about service quality management.
4. Describe about service transition and operation.
5. Explain about service IT Services and Technology components in detail.

Apply:

1. Illustrate ITIL service lifecycle.
2. Illustrate PDCA Model to control and manage quality.
3. Exhibit two successful approaches for IT services.
4. Demonstrate proactive risk strategy with one example.
5. Exhibit the steps involved in service design.

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, 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:

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. Dr.D. Tamilselvi – dtamilselvi@tce.edu
3. Ms. A.Malini – amcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1I	1	0	-	1

C1I Introduction to Mainframe Systems 1: 0

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.

Program Outcomes addressed

a. Graduate will have knowledge about main frame systems and Enterprise level IT Infrastructure
b. Graduates will use RAS features of system z and its essential requirements for mission critical workloads
c. Students will have opportunities to familiarize system z Enterprise computing and information technology and develop applications for large IT organizations

Competencies

At the end of the course the student should be able to

1. Identify the Main Frame System z basics and its components
2. Understand the TSO/ISPF, SDSF, JCL
3. Understand the various functionalities associated with deploying a complex application
4. Understand the DB2, WAS and REXX commands

Assessment Pattern

	Bloom's Category	Terminal Examination
1	Remember	10
2	Understand	30
3	Apply	60

4	Analyze	0
5	Evaluate	0
6	Create	0

Course Level Learning Objectives:

Remember:

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.

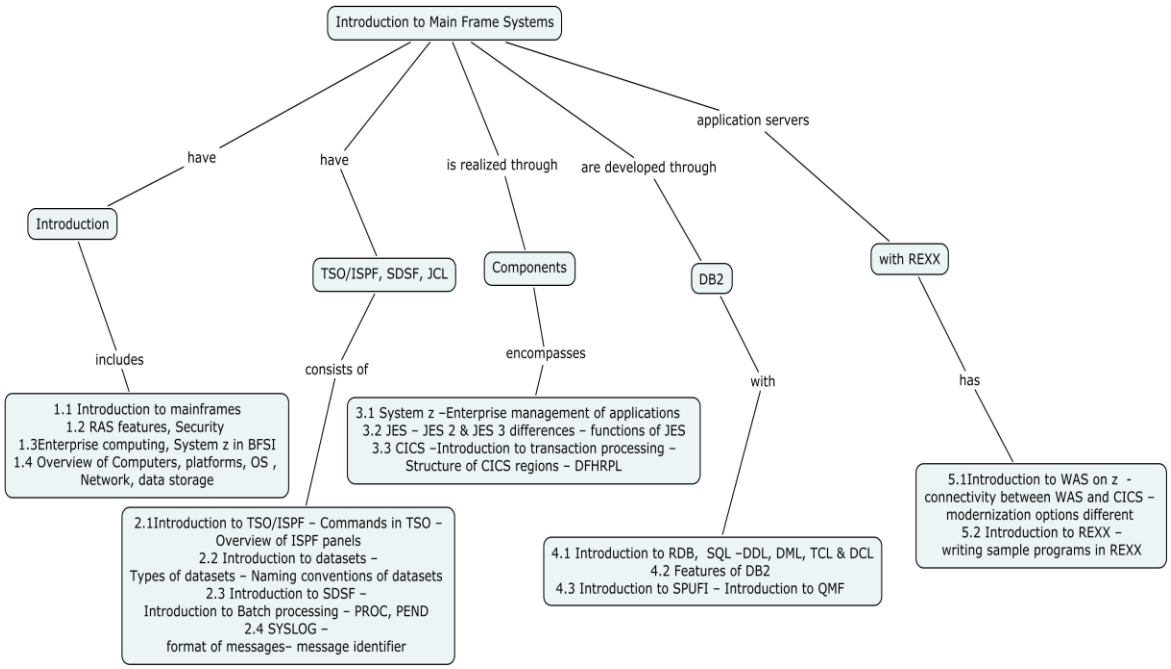
Understand:

1. Describe the requirements of System z in BFSI, aviation and other sectors.
2. Outline the Overview of ISPF panels.
3. Explain the structure of CICS regions.
4. Explain the features of DB2.
5. Write about Horizontal clustering & vertical clustering?

Apply:

1. Illustrate TSO Commands.
2. Illustrate JCL Commands.
3. Explain the functions of JES.
4. Explain the features of WAS on z?
5. 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

References:

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	1

	programs in REXX	
	Total	**16**

Course Designers:

1. Mr. K.Shreekanth - *kshreeka@in.ibm.com*
2. Dr.D. Tamilselvi – dtamilselvi@tce.edu
3. Mr.S. Vignaraj – vignaraj@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1J	1	-	--	1

C1J Practical approaches to Networking 1:0

(Common to CSE, IT and ECE Departments)

Preamble: This course is offered as a 1 credit elective to the IT related branches. 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.

Program Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.
b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.
c. Graduate will demonstrate skills to use modern engineering tools, software's and equipment to analyze problems

Competencies

At the end of the course the student should be able to

1. Understand and explain implementations of DNS and DHCP on a windows server
2. Understand and explain RPC
3. Get familiar with using network packet analysis tools, to understand, identify and troubleshoot network related issues
4. Demonstrate a deep understanding of the common networking services like DNS and DHCP
5. Implementation of common networking services in the labs
6. Get familiar with protocols like Microsoft RPC
7. Get familiar with usage of network packet analyzing tools like netmon. They will use these tools to understand the working of several protocols, by looking at the packets in the wire.

Assessment Pattern

	Bloom's Category	End-semester examination
1	Remember	20

2	Understand	30
3	Apply	50
4	Analyze	0
5	Evaluate	0
6	Create	0

Course Level Learning Objectives

Remember:

1. State the need for DHCP.
2. What is a DNS resource record?
3. List the various options of a DHCP scope.
4. Define RPC.

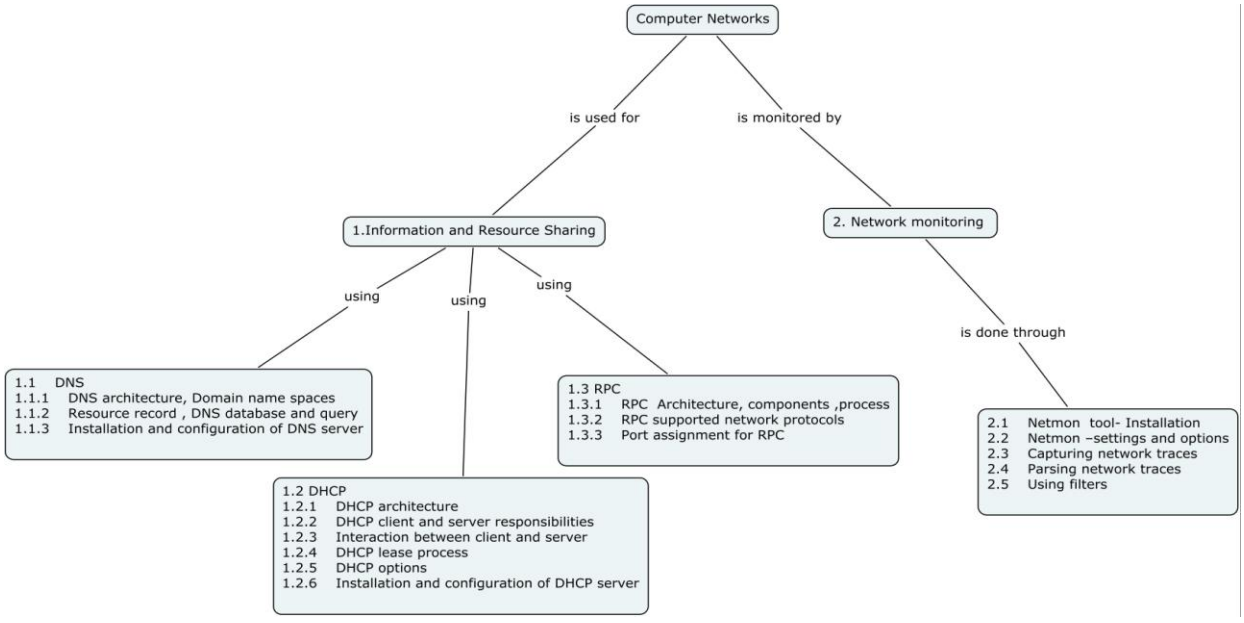
Understand:

1. List the different DNS domain names.
2. Draw the DHCP architecture.
3. State the different stages in DHCP lease process.
4. List the different components of RPC.

Apply

1. Install and implement DNS server in your network.
2. Install and implement DHCP server in your network.
3. Using Netmon tool capture one sequence of TCP/IP handshake.
4. Capture a sequence of DNS request and response to any website accessed.

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 Lectures 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. Shashi Kumar G shashikg@microsoft.com

2.C.Senthilkumar cskcse@tce.edu

3. T.Manikandan tmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
C1K	1	-	-	1

C1K Introduction to Microsoft ADDS 1:0

(Common to CSE, IT and ECE Departments)

Preamble:

Active Directory Domain Services (AD DS) is Microsoft's implementation of a directory service that provides centralized authentication and authorization services. AD DS in Windows Server 2008 provides a powerful directory service to centrally store and manage security principals, such as users, groups, and computers, and it offers centralized and secure access to network resources.

This course aims at exploring the features of active directory domain services and the logical and physical administration of active directory. An overview of backup and restore services of active directory is also provided.

Prerequisite C44**-**Operating Systems: Principles and Design C54**-**Computer Networks: Principles

Program Outcomes addressed

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

e. Graduates will use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

1. Describe and configure server roles with Active Directory Services in Windows Server 2008.
2. Plan for and deploy Active Directory Domain Services.
3. Manage accounts, subnets, Site-Links, Group Policy, and DNS configuration with Active Directory Domain Services.
4. Able to explore the physical components of Active Directory.
5. Use backup and restore features in Active Directory Domain Services.

Assessment Pattern

	Bloom's Category	Terminal Examination
1	Remember	30
2	Understand	40
3	Apply	30
4	Analyze	0
5	Evaluate	0
6	Create	0

Course Level Learning Objectives

Remember:

1. What is Microsoft Active Directory?
2. What is LDAP?
3. What do you mean by group policy?
4. Define: subnets and connectors.

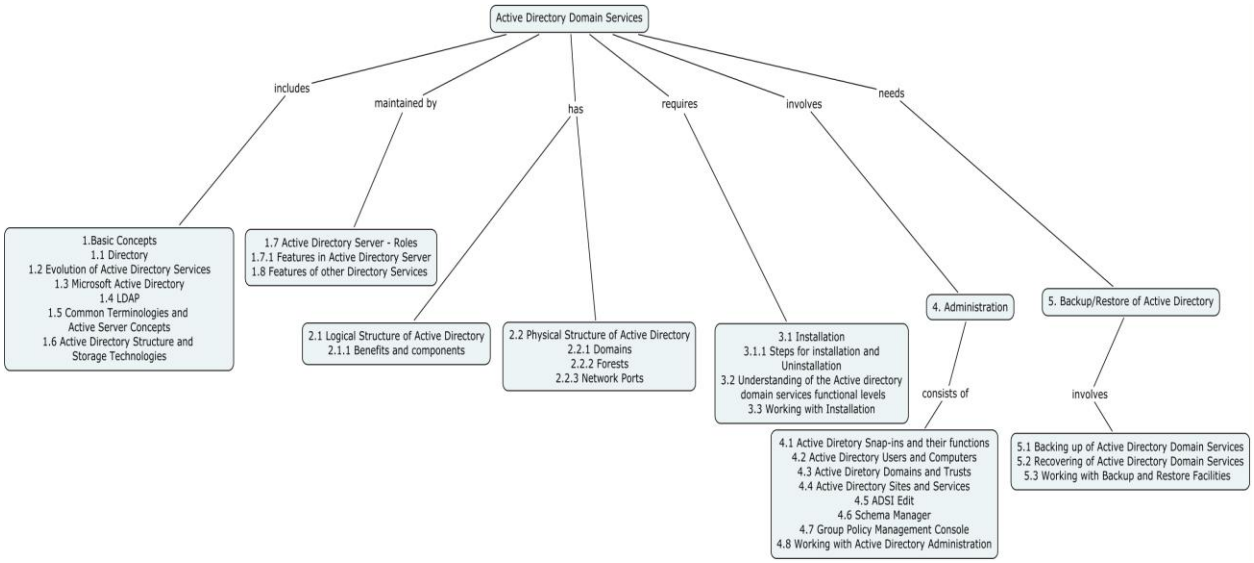
Understand:

1. Explain the features of active directory domain server.
2. How do you manage various computer accounts under the active directory service?
3. How active directory supports restoring facility? Illustrate with an example.
4. What is meant by replication? How it is implemented in active directory?

Apply

1. Illustrate the steps involved in installing an Active directory domain services.
2. How to create users and organizational units in active directory?
3. How backup of active directory is done in windows server 2008? Illustrate with an example.

Concept Map



Course Contents and Lectures schedule

1.	**Introduction to Microsoft Active Directory**	
1.1	Basic Concepts and Directory	1
1.2	Evolution of Active Directory Services	
1.3	Microsoft Active Directory	
1.4	LDAP	
1.5	Common Terminologies and Active Server Concepts	2
1.6	Active Directory Structure and Storage Technologies	1
1.7	Roles of Active Directory Server	1
1.7.1	Features in Active Directory Server	
1.8	Features of other Directory Services	
2	**Active Directory Domains and Forests**	
2.1	Logical Structure of Active Directory	

2.1.1	Benefits and components	1
2.2	Physical Structure of Active Directory	3
2.2.1	Domains	
2.2.2	Forests	
2.2.3	Network Ports	1
3	**Installation of Active Directory Domain Services in Windows 2008 R2**	
3.1	Requirements of Installation	1
3.1.1	Steps for installation and Uninstallation	
3.2	Understanding of the Active directory domain services functional levels	1
3.3	Working with Installation	2
4	**Active Directory Administration**	
4.1	Active Directory Snap-ins and their functions	1
4.2	Active Directory Users and Computers	2
4.3	Active Directory Domains and Trusts,	
4.4	Active Directory Sites and Services,	
4.5	ADSI Edit	2
4.6	Schema Manager	
4.7	Group Policy Management Console	
4.8	Working with Active Directory Administration	1
5	**Backup/Restore of Active Directory**	
5.1	Backing up of Active Directory Domain Services	1
5.2	Recovering of Active Directory Domain	1

	Services	
5.3	Working with Backup and Restore Facilities	1
	TOTAL	**23**

Syllabus:

Introduction to Microsoft Active 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:

http://technet.microsoft.com/en-us/library/cc731053(WS.10).aspx – Active Directory Domain Services Overview
http://technet.microsoft.com/en-us/library/cc773108(v=WS.10).aspx – Operations Master Roles
http://technet.microsoft.com/en-us/library/cc759186(v=WS.10).aspx - Active Directory Structure and Storage Technologies
http://technet.microsoft.com/en-us/library/cc754697.aspx - Understanding Sites, Subnets and Site Links

http://technet.microsoft.com/en-us/library/cc759073(v=ws.10).aspx
http://technet.microsoft.com/en-us/library/cc783351(v=ws.10).aspx#w2k3tr_logic_how_rqma

http://technet.microsoft.com/en-us/library/cc771433(v=ws.10).aspx - Scenarios for Installing AD DS

http://technet.microsoft.com/en-us/library/cc771188(v=ws.10).aspx - Requirements for installing AD DS
http://technet.microsoft.com/en-us/library/cc772464(v=ws.10).aspx - Insalling a New Forest
http://technet.microsoft.com/en-us/library/understanding-active-directory-functional-levels(v=ws.10).aspx - Understanding Active Directory Domain Services Functional Levels

http://technet.microsoft.com/en-us/library/cc753298.aspx - Group Policy Management Console
http://technet.microsoft.com/en-us/library/cc816584(v=WS.10).aspx - Backing Up Active Directory Domain Services
http://technet.microsoft.com/en-us/library/cc816751(v=WS.10).aspx - Recovering Active Directory Domain Services

http://technet.microsoft.com/en-us/library/cc794908(v=ws.10).aspx - Administering Active Directory Domain Services
http://technet.microsoft.com/en-us/library/cc754217.aspx - Active Directory Users and Computers
http://technet.microsoft.com/en-us/library/cc770299.aspx - Active Directory Domains and Trusts
http://technet.microsoft.com/en-us/library/cc730868.aspx - Active Directory Sites and Service
http://technet.microsoft.com/en-us/library/cc730667.aspx - Schema Manager

Course Designers:

1. Mr. G. Shashi Kumar, Escalation Engineer, Microsoft GTSC, Bangalore, shashikg@microsoft.com
2. Dr.C.Deisy, cdcse@tce.edu
3. Mrs.B.Subbulakshmi bscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGA	3	0	-	3

CGA ESSENTIALS OF INFORMATION TECHNOLOGY 3:0

Preamble:

This course is offered in the sixth semester for the Non Computer Science students. As prerequisites, the students must have the knowledge of basics of operating systems, computer organization and design and data structures in the previous semesters. This course gives the non-cse students an industry-oriented knowledge of software development.

Program Outcomes addressed:

a. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

b. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

c. Graduates will demonstrate an ability to engage in life-long learning.

d. Graduates will demonstrate an ability to use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

e. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, mathematics and science

Competencies:

Student will be able to

1. Identify a suitable configuration of computer system for any application.
2. Develop software as per the appropriate software life cycle model.
3. Organize and maintain the information of an organization.
4. Analyse the complexity of the algorithm.
5. Design a suitable algorithm for a given problem

Assessment Pattern:

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	40	20	20
2	Understand	30	40	30
3	Apply	30	40	50
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. What is a computer system?
2. Give an example for tertiary storage device.
3. Give the worst case for merge sort
4. List down the drawbacks of traditional file processing system?
5. What is a Knapsack problem?
6. Define a product.
7. List down the phases in Waterfall model.
8. Write down steps that take place in performing an I/O operation?
9. Draw the structure of memory hierarchy.
10. List down the functional units of a computer system.

Understand

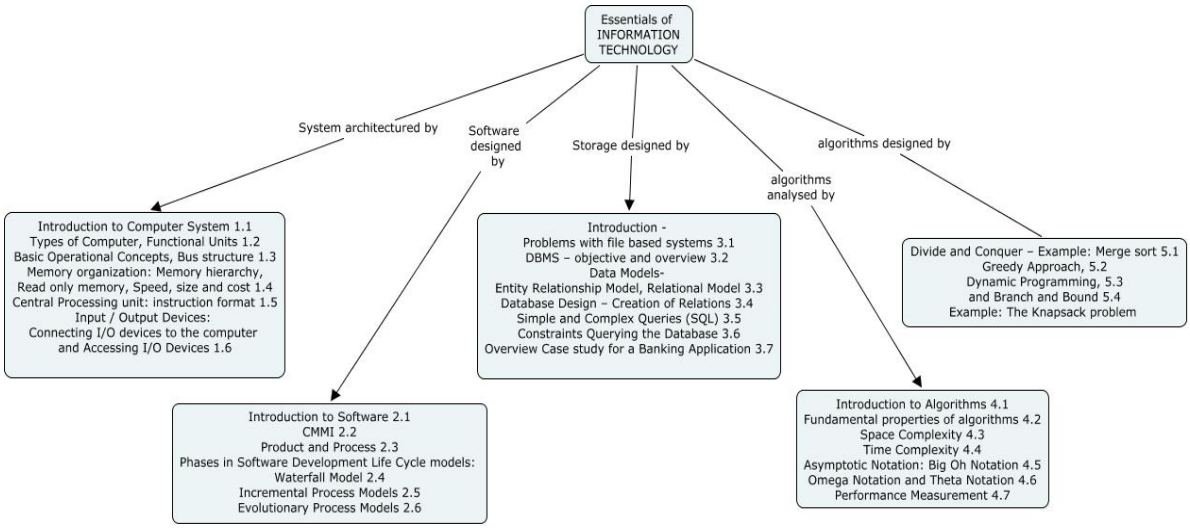
1. What is the purpose of DBMS?
2. Compare the speed, size and cost factor of primary and secondary storage device.
3. How product is related to process?
4. Describe the phases in software development life cycle.
5. How the software is tested?
6. How a weak entity set is represented in E-R Diagram?
7. Compare and contrast Dynamic programming and Greedy Approach.
8. What are the three different asymptotic notations?
9. How the running time of an algorithm is calculated?
10. What are the constraints that are used to represent the relationship between the relations?

Apply

1. Represent the expression X=A+B in instructions using
    a. One address format
    b. Two address format
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. Find the big-oh (O) notation for the function f (n) $=6n^2 + 135$?
5. Design an ER diagram for a University.
6. Calculate the time complexity for Merge Sort.
7. Give an efficient solution for knapsack problem.

8. To achieve rapid development the RAD model assumes the existence of one thing. What is it and why is the assumption not always true?

Concept map:



Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Computer Hardware and System Software**	
1.1	Introduction to Computer System	1
1.2	Types of Computer, Functional Units	1
1.3	Basic Operational Concepts, Bus structure	2
1.4	Memory organization: Memory hierarchy, Read only memory, Speed, size and cost	2
1.5	Central Processing unit: Instruction format	2
1.6	Input / Output Devices: Connecting I/O devices to the computer and Accessing I/O Devices	1
2	**Software development Life Cycle**	
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	3
3	**Relational Data base management systems**	
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	2
3.4	Database Design – Creation of Relations	2
3.5	Simple and Complex Queries (SQL)	2
3.6	Constraints Querying the Database	2
3.7	Overview Case study for a Banking Application	2
4	**Analysis of algorithms**	
4.1	Introduction to Algorithms	1
4.2	Fundamental properties of algorithms	1
4.3	Space Complexity	1
4.4	Time Complexity	2
4.5	Asymptotic Notation: Big Oh Notation	2
4.6	Omega Notation and Theta Notation	2
4.7	Performance Measurement	2
5	**Design of algorithms**	
5.1	Divide and Conquer – Example: Merge sort	2
5.2	Greedy Approach - Example: The Knapsack problem	2
5.3	Dynamic Programming - Example: The Knapsack problem	2
5.4	Branch and Bound – Example: The Knapsack problem	2
	Total No of Hours	50

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.

Analysis of algorithms - Introduction to Algorithms- Fundamental properties of algorithms – Space Complexity - Time Complexity - Asymptotic Notation: Big Oh Notation, Omega Notation, and Theta Notation - Performance Measurement.

Design of Algorithms - Divide and Conquer – Example: Merge sort - Greedy Approach, Dynamic Programming and Branch and Bound: Example - The Knapsack problem.

Text Book:

1. Computer Organization & Architecture (Sixth Edition) By William Stallings (PHI)
2. Computer Organization (Fifth Edition) By Carl Hamacher, Zvonko Vranesic & SafwatZaky (TMH).
3. M. Morris Mano, Computer System Architetutre, PHI 2003
4. J.P. Hayes, Computer Architecture, McGraw-Hill 2002 Sivarama. P. Dandamudi,
5. Roger S. Pressman , Software Engineering: A Practitioner's Approach (5th edition), Mc Graw Hill 2000[Chapters 1,2,17,18]
6. Henry F Korth, Abraham Silberschatz, Database System Concept, 5th ed. McGraw-Hill International editions.
7. Ragu Ramakrishnan, Database Management Systems, McGraw Hill Edition, 1998.
8. Ramez Elmasri and Samrath Navetha, Fundamentals of Database System, Addison Werely, 1994
9. Alfred V.Aho Hopcraft, Ullman, Data Structures and Algorithms, Addison Wesley publications ,2000
10. Ellis Horowitz ,Sartaj Sahni, Fundamentals of Computer Algorithms, Galgotia Publications,2002.
11. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, McGraw Hill
12. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India, 2003.

Course Designers:

1. G. Madhu Priya gmadhupriya@tce.edu

1. M. Suguna mscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGB	3	0	-	3

CGB SOFTWARE DEVELOPMENT **3:0**

(Common to Non CSE and Non IT)

Preamble: The course aims at exploring the development of web applications featuring the Internet, covering aspects ranging from concepts of OOAD,Client server architecture , protocols ,web server and web browser. An overview of user interface design is also provided.

Program Outcomes addressed

a.An ability to identify, formulate and solve engineering problems.

b.An ability to analyze and design a system or component, or process to meet stated specifications.

c.An ability to be able to communicate effectively in both verbal and written form.

Competencies

At the end of the course the student will be able to

1. Prepare input-output-functional specifications of an information processing system in the object oriented framework
2. Develop use-case model, activity diagram, interaction diagram and class diagram from the user's requirements using UML.
3. Structure the given information system in terms view layer (interface), business layer and access layer (communication with database).
4. Select an appropriate approach to the identification of classes and their behaviors for a given problem.
5. Design access layer for communication with data and the view layer for mapping UI objects to the view layer objects.
6. Design User Interfaces as per the user requirements

Assessment Pattern:

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	30	40
2	Understand	30	20	20
3	Apply	20	20	10
4	Analyze	10	20	20
5	Evaluate	10	10	10
6	Create	0	0	0

Remember:

1. Explain the characteristics and features of object-oriented philosophy, with regards to objects, classes.
2. What is Object Oriented Development Life Cycle? What are the different methodologies in Object Oriented System Development?
3. What is Unified Approach? What is UML? How are the various models represented in UML?

1a. List the advantages of Object Oriented Development?(factual)
1b. How does Object Oriented Development differs from traditional development.(Factual, Conceptual)
1c. What is polymorphism, inheritance? (Factual)
1d. What is meant by the term object orientation? (Factual)
1e. Write a note on object and classes. what is the meaning of the terms object identity and persistence.(Factual)
1f. Throw some light on the principles of encaptulation and information hiding. (Factual)
2a. What are the various phases of Software development Process? (Factual, Procedural)
2b. Write a note on Rapid Application Development .(Factual)
2c. What are the various quality measures that are taken into consideration for software evaluation? Explain with a diagram. (factual, Conceptual)
2d. Briefly Describe the Booch System Development processes. (factual, conceptual)
3a. What are various processes and components of the unified approach.(factual, conceptual)
3b. What are the advantages of UML? (conceptual)
3c. What are the various phases of OMT? Briefly describe each phase. (Factual, Procedural)

Understand:

1. Convert the user requirement to model an object oriented systems.
2. Categorize the various approaches to model on Object Oriented System

1a. Differentiate the user needs for an ATM system to independent classes.(conceptual and procedural)
1b. Represent the functions of an ATM in an sequential diagram(procedural)
2a. Summarize the advantages and disadvantages of Layered approach

(factual, conceptual).

2b.Distinguish the waterfall approach with prototyping.(Conceptual , Procedural)

2c. Compare and contrast Unified Approach with other traditional approaches.

Apply:

1.Application of various principles, approaches, axioms to develop systems

1a. Apply the Waterfall model approach to develop an ATM system. (Conceptual, Procedural)

1b. Use Object Oriented approach to construct a model of an ATM system. (Conceptual, Procedural)

1c. Construct Sequence /Collaboration Diagrams for the deposit checking use case of the Bank system. (Conceptual, Procedural)

1d. Develop a Use Case Diagram that demonstrates the 'extend' and the Uses relationship for a library information system. (Conceptual, Procedural)

1e. Apply noun phrase approach for identifying classes in Library Management system. (Procedural)

1f. Demonstrate with an example the use of a sequence /collaboration diagram in identifying classes. (Procedural)

1g. Use the Noun Phrase Approach to identify classes in Library Management system. (Procedural)

1h. Using the UI Design rules design an interface for a user who is completely new to the computer and needs to manage a superstore. (Factual, Procedural)

1i. Taking a sample case study Explain the object oriented systems development by use case driven approach.(Procedural)

Analyze:

1.Compare and contrast the various Object oriented methodologies.

2.Comparison between various concepts and approaches in OOAD

3.Choice of different diagrams and models in UML

1a. What are the strengths of Jacobson methodology over Booch methodology? (conceptual, Procedural)

2a. Differentiate between a pattern and framework.(Procedural)

2b. You are required to design a management system for a hospital which will automate the hospital .which approach will you suggest given a choice between the Prototyping and waterfall approach for development of the system? Justify. (Conceptual)

3a. What are collaboration and sequence diagrams. How would you make a choice regarding their use? Explain with an example. (Factual, Conceptual, Procedural)

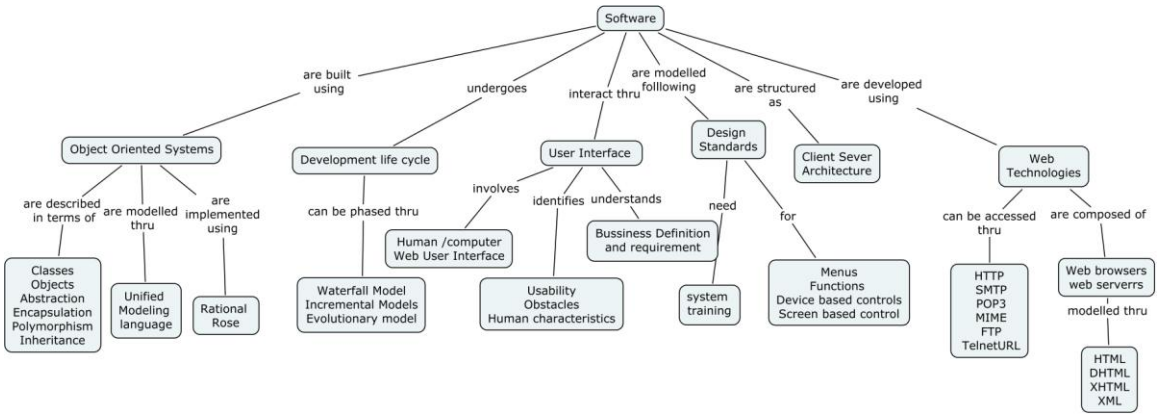
Evaluate:

1. Evaluation and Justification of an approach for a given application.

1a. Select an approach between the Prototyping and waterfall approach for development of object oriented system? Justify.(Conceptual,procedural)

1b. Outline the coherencies between Unified approach and traditional Methodologies(factual,conceptual)

Concept Map:



Course Contents :

No.	Topic	No. of Lectures
1.	**Object Oriented concepts**	
1.1	Classes	1
1.2	Objects	
1.3	Polymorphism	1
1.4	Inheritance	1
1.5	Relationships	2
1.6	Unified Modelling language	2
1.7	Introduction to rational rose	3
2	**Software development Life Cycle**	
2.1	Introduction to Software	1

No.	Topic	No. of Lectures
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	3
3.	**User Interface Design**	
3.1	Human-Computer interface	1
3.2	web user interface	2
3.3	User interface design process	2
3.4	human characteristics in design Human interaction speed	1
3.5	Business Definition and Requirements Analysis	2
3.6	Determining Basic business functions-	2
3.7	**Design standards** - System training	1
3.8	Human consideration in screen design	2
3.9	Presentation Styles– Types Controls	2
3.10	Control Screen Design	2
4	**Client/Server Architecture**	
4.1	Building blocks of the client/server architecture,	2
4.2	Characteristics of the Client/Server architecture Classifying Client/Server Architecture	2
4.3	**Developing Client Server Applications**	2
4.4	Development Tools Role of middleware in client/server communication	2
5.	**Web Technologies**	
5.1	Internet and World Wide Web Protocols	2
5.2	Components of web application Browsers and Web Servers	1

No.	Topic	No. of Lectures
5.3	Scripting Application servers	2
	Total No of Hours	50

Syllabus

Object Oriented concepts Classes -Objects -Polymorphism -Inheritance Relationships - Unified Modelling language - Introduction to rational rose- **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.**User Interface Design** Human-Computer interface web user interface User interface design process human characteristics in design Human interaction speed Business Definition and Requirements Analysis Determining Basic business functions- **Design standards** System training Human consideration in screen design Presentation Styles– Types Controls Control Screen Design **Client/Server Architecture** Building blocks of the client/server architecture, Characteristics of the Client/Server architecture Classifying Client/Server Architecture **Developing Client Server Applications** Development Tools Role of middleware in client/server communication **Web Technologies** Internet and World Wide Web Protocols Components of web application Browsers and Web Servers Scripting Application servers

Textbook

1. Wilbent. O. Galitz ,"The Essential Guide to User Interface Design", John Wiley& Sons, 2001.
2. Alan Cooper, "The Essential Of User Interface Design", Wiley – Dream Tech Ltd., 2002.
3. Robert Orfali, Dan Harkey and Jerri Edwards, Essential Client / Server Survial Guide, John Wiley and sons Inc..
4. Bvar, B.H., Implementing Client / Server Computing : A Strategic Prospective, McGraw Hill, 1993.
5. Bruce Elbert, Client / Server Computing, Artech Press, 1994.

Reference Books

1. Alex Berson, Client / Server Architecture, McGraw Hill, 1996
2. Bjarne Stroustrup, 'The C++ Programming Language', Third Edition, Addison-Wesley, 2000

3. Martin fowler, 'UML distilled: a brief guide to the standard object modeling language', Addison-Wesley, Third Edition, 2003
4. Wilbent. O. Galitz ,"The Essential Guide to User Interface Design",John Wiley& Sons, 2001.

COURSE DESIGNERS:

1. S.Prasanna sprcse@tce.edu
2. C.SenthilKumar cskcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGC	2	1	-	3

CGC DESKTOP PUBLISHING
(Common to Non CSE and Non IT) 2:1

Preamble: This course is a study of desktop publishing and the current state of the desktop publishing industry. Emphasis will be placed on how to actually create a publication from start to finish and include emphasis on types of publications

Program Outcome addressed:

a. Graduates will gain a working knowledge of the desktop publishing process

b. Graduates will acquire at least a minimum level of competency in areas such as text formatting, Presentation slide preparation and in layout design.

c. Graduates will be capable of creating publications useful in the classroom environment for both student use and adult communications

Competencies:

1. Student will be capable of specifying hardware and software requirements for DTP
2. capable of preparing a well formatted document.
3. Become familiar in preparing a customized and animated good presentation slides
4. capable of designing a front page (for a news paper).
5. capable of designing a front-page for a bulletin

Assessment Pattern:

Sl.No	Bloom's Category	Test1	Test2	Test3/End Semester Examination
1	Remember	20	20	20
2.	Understand	20	20	20
3.	Apply	20	20	20
4.	Analyze	0	0	0
5.	Evaluate	0	0	0
6.	Create	40	40	40

Course Level Learning Objectives

Remember:

1. Define DTP.
2. List out the Hardware and Software requirements for DTP

3. Narrate the symbols used for formatting a text document.
4. What are the various options available to customize the design templates?
5. State the need for effective front page designing of a magazine.
6. What are the special features available in the Excel tool bar?

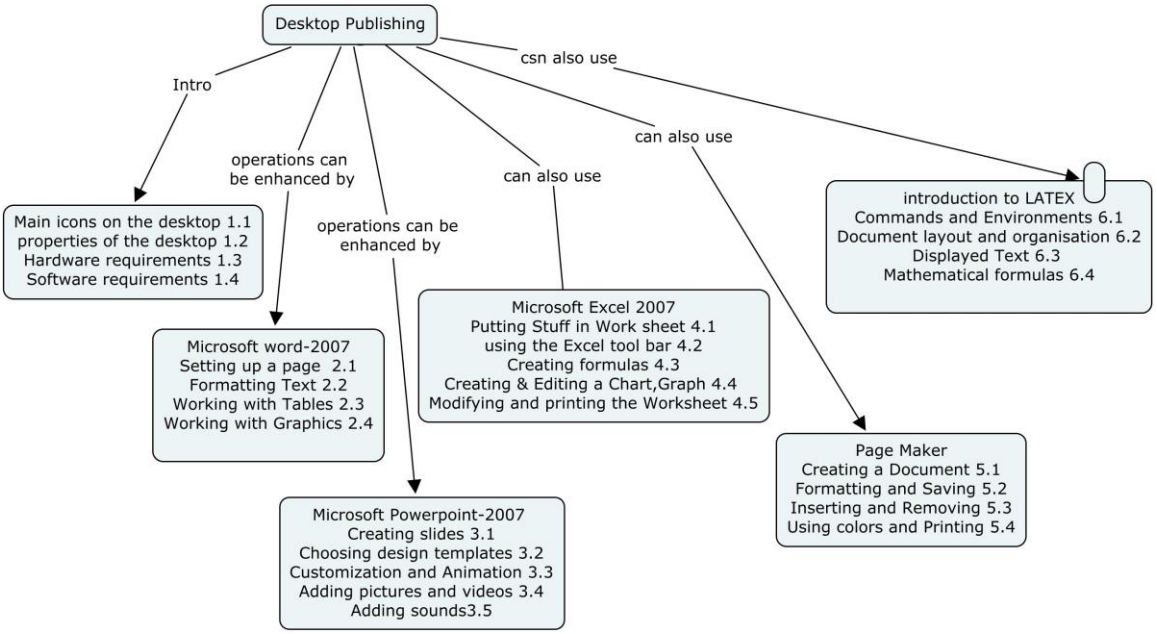
Understand:

1. What are the procedures adopted for page setup of a text document.
2. Narrate the presentation sequence to be followed for a project presentation.
3. Explain the procedure involved in the design of a front page.
4. How cell alignment is carried our in Excel?
5. Explain the steps involved in incorporating special effects in LATEX.

Apply

1. Create a project document for with necessary page setup, Margins, Font size and Font type. See that every page should contain twenty-five lines in double spacing.
2. Create a PowerPoint presentation for a project including necessary background, adding headers and footers, inserting page numbers and custom animated..
3. Make use of the hyperlink option available so that two applications can be utilized (Power point and MS word).
4. Apply the formulas in appropriate place in Excel to find the average of the given data.
5. Design a page for a bulletin using page maker.
6. Use the LATEX to create a document.

Concept Map



Course content and lecture schedule

Sl.No	Topics	No of periods
1.	**Introduction to Desktop publishing**	1
1.1	Main icons on the desktop	2
1.2	Properties of the desktop	2
1.3	Hardware requirements	2
1.4	Software requirements	2
2	**Microsoft Word 2007**	
2.1	Setting up a page	2
2.2	Formatting text	2
2.3	Working with tables and Styles	2
2.4	Working with graphics	2
3	**Microsoft Power point-2007**	
3.1	Creating slides	1
3.2	Choosing design Templates	1
3.3	Customization and Animation	2
3.1	Adding pictures and videos	2
3.5	Adding sounds	1
4.	**Microsoft Excel -2007**	
4.1	Putting stuff in a worksheet	1
4.2	Using the Excel Tool bars,	1
4.3	Creating formulas	2
4.4	Creating and Editing a Chart, Graph	2
4.5	Modifying and printing the worksheets	2
5	**Page Maker**	2
5.1	Creating a new document	2
5.2	Formatting and saving a document	2
5.3	Inserting and removing pages	2
5.4	Using colors and printing a document	2
6	**Introduction to LATEX**	
6.1	Commands and Environments	2
6.2	Document Layout and Organizations	2
6.3	Displayed Text	2
6.4	Mathematical formulas	2

Syllabus:

Introduction: Hardware and Software requirements for Desktop publishing, Introduction to the Desktop, Main Icons on the Desktop, Properties of the Desktop**,: Microsoft Word -2007,** Setting up a page**,** formatting text**,** working with tables and graphics **Microsoft PowerPoint- 2007**,Creating slides, Choosing design templates, Customization and Animation, Adding pictures, videos and sounds. **Microsoft Excel-2007,** Putting stuff in a worksheet, Using the Excel Tool bars, Creating formulas, Creating and Editing a Chart, Modifying and printing the worksheets. **Page Maker,** creating a new document, formatting, saving, inserting and removing pages, Using color and printing. **Introduction to LATEX,** Commands and Environments, Document Layout and Organizations, Displayed Text, Mathematical formulas

Text Books:

1. DTP Course 9th revised edition, Asian Computech Book, Computech Publications 2007
2. Word 2007 in Simple Steps by Cogent Solutions Inc, Dreamtech press, 2008
3. Working in MS OFFICE, Ron Mansfield, Tata McGraw Hill edition 2007
4. "Adobe PageMaker– Illustrated" by Kevin G.proot, 1st edition
5. A Guide to LATEX, Helmut Kopka and Patrick W.Daly, Addison Wesley Publications, 3rd edition, 2008

References:

1. DTP Revolutionary three stage self learning system Course kit -Vikas Gupta
2."Special edition using Microsoft office 2000 " Edbott and woody leonhard,Que's Special edition wopr pack.Tim j.walker
3."Microsoft office 2000 An introductory concept and techniques" Gary B shelly,Thomas J cashman ,Misty.E.vermat jan 2000 .Shelly cashman series.
4. MS Word 2007 , Vishnu Priya singh, Computech Publications, 2007.

Course Designers:

1. R.Chellamani. rcmcse@tce.edu
2. N.Balaji. nbcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGD	3	0	-	3

CGD Fundamentals of Cloud Computing 3:0

Preamble: This course is offered as a general elective to the non computer science under graduate students. It is aimed at introducing cloud computing and the various services offered by the cloud providers. By this course the student could understand that without any infrastructure investments, one can get powerful software and massive computing resources quickly with lower up-front costs and fewer management headaches by moving to cloud computing.

Programme Outcomes addressed

b. Graduates will identify, formulate and solve engineering problems

e. Graduates will use techniques, skills and modern engineering tools to implement and organize engineering works under given constraints

Competencies

At the end of the Course, the student will be able to

1. Explain the evolution of cloud computing and the various web services offered.
2. Illustrate the ways to build the cloud infrastructure
3. Illustrate the different ways to access the cloud services.
4. Explain the various cloud service providers and their offered services.
5. Illustrate the role and benefits of virtualization in cloud computing.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	30	20	20	20
2	Understand	40	40	20	20
3	Apply	30	40	60	60
4	Analyze	0	0	0	0
5	Evaluate	0	0	0	0
6	Create	0	0	0	0

Course Level Learning Objectives

Remember

1. List the benefits of Cloud Computing.
2. Define optimized real time routing
3. Define a web application framework.
4. Define Platform-as-a-Service.
5. List some of the tools that can be used for virtualization.

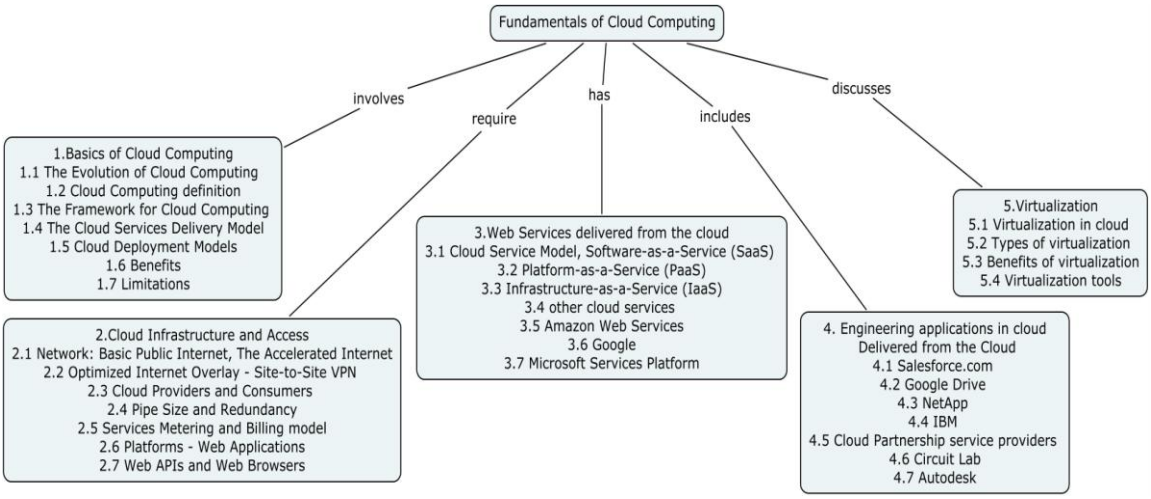
Understand

1. Describe the evolution of Cloud Computing.
2. Compare Software-as-a-Service with Platform-as-a-Service
3. Explain the working of an API with an example.
4. Discuss on Infrastructure as a Service offering.
5. Explain any one type of virtualization.

Apply

1. Illustrate the roots in the evolution of cloud computing.
2. Summarize the pros and cons of various cloud deployment models.
3. Illustrate the various services offered by Microsoft service provider.
4. Demonstrate the steps to construct the virtual machine using the virtualization tool.

Concept Map:



Syllabus

Basics of Cloud Computing: The Evolution of Cloud Computing - Cloud Computing definition - The Framework for Cloud Computing - The Cloud Services Delivery Model - Cloud Deployment Models – Benefits – Limitations **Cloud Infrastructure and Access:** Network: Basic Public Internet - The Accelerated Internet - Optimized Internet Overlay - Site-to-Site VPN - Cloud Providers and Consumers - Pipe Size - Redundancy, Services – metering and billing models – Platforms - Web Applications - Web APIs and Web Browsers **Web Services delivered from the Cloud :** Cloud Services Model - Software-as-a-Service (SaaS) - Platform-as-a-Service (PaaS) - Infrastructure-as-a-Service (IaaS) and other services, Case Study - Amazon Web Services - Google - Microsoft Services Platform **Engineering applications in cloud:** Salesforce.com – Google Drive – NetApp – IBM - Cloud Partnership service providers viz. Circuit Lab, Autodesk **Virtualization:** Virtualization in cloud – types of virtualization – benefits of virtualization – virtualization tools

References:

1. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", McGraw-Hill 2010.
2. John Rittinghouse, James Ransome, "Cloud Computing: Implementation, Management and Security", CRC Press 2010.
3. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, "Cloud Computing: Principles and Paradigms", John Wiley & Sons, 2011.
4. http://drive.google.com/start
5. https://www.circuitlab.com/
6. https://360.autodesk.com/landing
7. http://www.salesforce.com/service-cloud/overview/

Course contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Basics of Cloud Computing (9)**	
1.1	The Evolution of Cloud Computing	2
1.2	Cloud Computing definition	1
1.3	The Framework for Cloud Computing	1
1.4	The Cloud Services Delivery Model	1
1.5	Cloud Deployment Models	2
1.6	Benefits	1
1.7	Limitations	1
2	**Cloud Infrastructure and Access (9)**	
2.1	Network: Basic Public Internet, The Accelerated Internet	1
2.2	Optimized Internet Overlay - Site-to-Site VPN	2
2.3	Cloud Providers and Consumers	1
2.4	Pipe Size and Redundancy	1
2.5	Services Metering and Billing model	2
2.6	Platforms - Web Applications	1
2.7	Web APIs and Web Browsers	1
3	**Web Services delivered from the Cloud (8)**	
3.1	Cloud Service Model, Software-as-a-Service (SaaS)	2
3.2	Platform-as-a-Service (PaaS)	1
3.3	Infrastructure-as-a-Service (IaaS)	1

3.4	other cloud services	1
3.5	Amazon Web Services	1
3.6	Google	1
3.7	Microsoft Services Platform	1
4	**Engineering applications in cloud (7)**	
4.1	Salesforce.com	1
4.2	Google Drive	1
4.3	NetApp	1
4.4	IBM	1
4.5	Cloud Partnership service providers	1
4.6	Circuit Lab	1
4.7	Autodesk	1
5	**Virtualization (7)**	
5.1	Virtualization in cloud	2
5.2	Types of virtualization	2
5.3	Benefits of virtualization	1
5.4	Virtualization tools	2
	Total No of Hours	40

Course Designers:

1. Mrs. G. MadhuPriya gmadhupriya@tce.edu
2. Mrs. J. Jane Rubel Angelina janerubel@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGE	3	-	-	3

CGE Web Technologies 3:0

Preamble: This course is offered in the Seventh semester for the Non CSE and IT students of Under Graduation. The students will learn how to represent the structure and designs using HTML and PHP and other related web technologies. The student gains understanding of how the internet application works and develops web programming skills. The course will establish a professional, client-based attitude towards web design.

Program Outcomes addressed

b. Graduates will able to use vendors support for HTML and PHP.

c. Graduates will learn how to do client side as well as server-side programming.

d. Graduates will create and develop interactive web pages on the World Wide Web.

Competencies

At the end of the course the student will be able to

1. Understand how the internet works with HTML and PHP.
2. Understand how the internet application works in general.
3. Construct HTML documents
4. Develop client/ server side programming for web application development.
5. Understand how to connect My SQL Databases with web pages

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	20	20	20	20
2	Understand	60	20	30	20
3	Apply	20	40	30	40
4	Analyze	0	0	0	0
5	Evaluate	0	0	0	0
6	Create	0	20	20	20

Course Level Learning Objectives:

Remember

1. Define IP Addressing
2. Define Computer Network.
3. List the types of Domain Name Space.
4. Define WWW
5. List the use of Client-Server Communication

6. List out the various web browsers that are commonly used.
7. Define: HTML
8. List out some of Basic Internet Protocols.
9. List out data types that are available in Web-Service.
10. Define CSS.
11. Define PHP.
12. Write the declaration of variable in PHP.

Understand

1. Sketch the Web Browser and Server Communication.
2. Write the Structure of HTML.
3. List out the version of HTML.
4. Write the basic document structure of PHP.
5. How does the internet work, brief with a sample application.
6. Write an HTML program to view student marks
7. Discuss in detail about HTML with an example.
8. Write an HTML program for user registration form
9. Explain the concept of CSS and its uses with an example.
10. Write an HTML program using PHP for library management.

Apply

1. Draw the HTML for Book Publisher along with List of Books.
2. Explain in detail about PHP Database Connectivity with an example of Student database.
3. Write a PHP program to display the message.
4. Write a PHP program to print the numbers from 0 to 50.
5. Write a PHP Script program to create table.
6. Explain in detail about Array with an example of Positive and negative numbers.
7. Write a PHP program to create user registration form.
8. Construct the book's HTML document using cascading style sheets.
9. Draw a picture describing the relationship between client/server objects used by My SQL
10. Write a PHP program for jewelry shop using CSS.

Create:

1. Create an HTML document that marks up your resume? (10)

    a. The first document will establish the **frameset**, which will be split into at least **three** frames. One of these frame windows must contain a document that provides navigation through this set of documents (a table of contents).
    b. The rest of the set will contain, at a minimum, **six** other documents, **four** of which must pass through an active frame. These may include documents that you have created for other assignments, or other pages you have created, but if, you choose to do this, you are responsible for the markup in these pages.
    c. One document in this set must contain a working **form** with at least **three** different types of **input elements**.

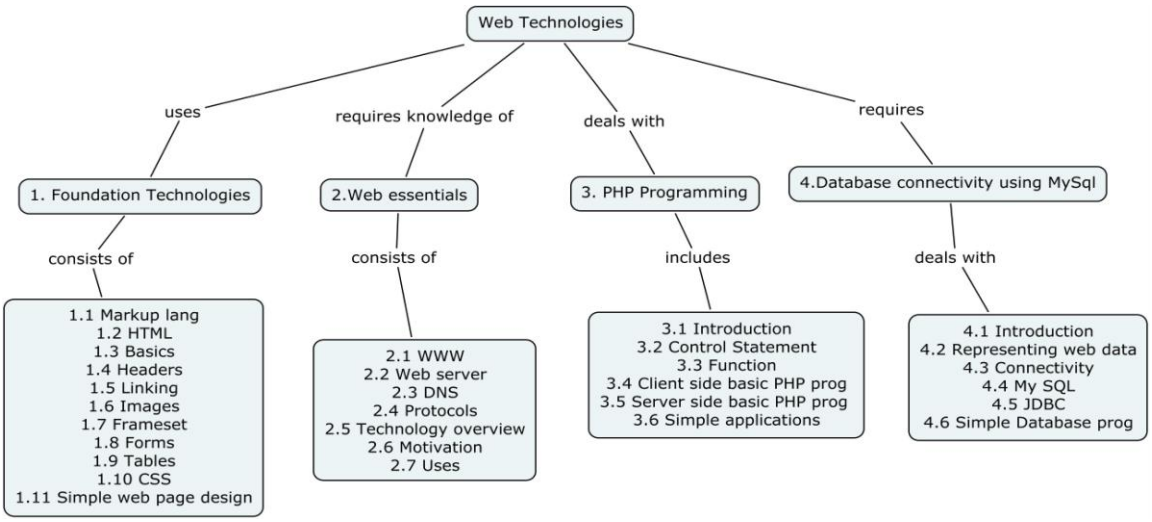
    It will also include:

    d. At least **two** external links to documents you do not own;
    e. At least **two** links to other documents you have created;
    f. At least **one** background color or image;

g. At least **three** images in the body of your resume pages; and
h. At least **two** targeted links between frames

2. Create a Web Page that applies the invert filter to all images if the user moves the mouse over it?
3. Create an HTML document that marks up your resume?
4. Create an HTML document titled "How to get Good Grades" Use <Meta> tags to include a series of keywords that describe your document.

Concept Map



Syllabus

Foundation Technologies – Markup language – HTML - Basics – Headers – Linking – Images – Frames –Frameset –Forms – Tables – CSS – simple web page designs. **Introduction to Web -** Web essentials – World Wide Web – Web server- DNS – Protocols- Technology overview - Motivation and Characteristics – uses – **PHP Programming –** Introduction to PHP – Control statements – Function – Basic PHP programs - Server side programming basics using PHP - simple web based applications - **Database Connectivity using My SQL-** Introduction to Database- Representing Web data – data base connectivity – Simple database programs.

References

1. Deitel and Deitel, "Internet and World Wide Web How to program", Prentice Hall of India, Fourth Edition, 2008.
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	**Foundation Technologies**	
1.1	Markup languages	1
1.2	HTML	1
1.3	Basics	1
1.4	Headers	1

1.5	Linking	1
1.6	Images	1
1.7	Frames - Framesets	2
1.8	Forms	2
1.9	Tables	1
1.10	CSS –Cascading style sheets	2
1.11	Simple web page designs	2
2	**Introduction to Web**	
2.1	Web essentials	1
2.2	World Wide Web	1
2.3	Web server	1
2.4	DNS – Protocols	1
2.5	Technology overview	1
2.6	Motivation and Characteristics	1
2.7	uses	1
3	**PHP Programming**	
3.1	Introduction to PHP	1
3.2	Control statements	1
3.3	Function	2
3.4	Client side programming basics using PHP	2
3.5	Server side programming basics using PHP	2
3.6	simple web based applications	2
4	**Database Connectivity using My SQL**	
4.1	Introduction to Database	1
4.2	Representing Web data	1
4.3	Data base connectivity	1
4.4	Introduction to Database	1
4.5	Representing Web data	1
4.6	Data base connectivity	1
4.7	Simple database programs	2
	Total	40

Course Designers:

1. Mr. T. Manikandan tmcse@tce.edu
2. Mrs. R.Suganya rsuganya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGF	3	0	-	3

CGF - Object Oriented Design 3:0

Preamble:

The Objective of this subject is to promote the practice of object oriented analysis and design concepts at a higher level of abstraction. This subject provides a sound understanding of the fundamental concepts and historical evolution of the object model, facilitate a mastery of the notation and process of object-oriented analysis and design and its realistic application.

Program me Outcomes addressed

a. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

a. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

1. Explain the basics of object oriented technology and models.
2. Knows how effectively use object-oriented technology to solve real problems.
3. To analyze a real world problem from requirement to implementation, using object-oriented analysis and design.
4. How to use the UML, diagrams through the various phases of software development lifecycle.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
1	Remember	20	20	20	10
2	Understand	50	40	30	30
3	Apply	0	40	20	45
4	Analyze	30	0	30	15
5	Evaluate	0	0	0	0
6	Create	0	0	0	0

Course Level Learning Objectives:

Remember

1. List the reasons for the complexity of software.

2. What is modeling? Write the various models in Object Oriented Development.
3. What are the different kinds of decomposition?
4. What is UML? List out its characteristics.
5. Define the term "Realization".
6. What do you understand from R 2.3.2 : s:= Student :: getDetail ("2KC51") ?
7. Define Swimlane with an example.

Understand

1. Explain in detail the structure of complex systems with suitable example.
2. Explain the evolution of object oriented model in detail.
3. Describe the various building blocks of UML.
4. Through which diagrams the static characteristics of the system can be shown? Explain the diagrams with a suitable example.
5. Explain how to represent the various relationships among the classes.
6. Distinguish Association and Link.

Apply

1. Consider a Car Company which sell both the used cars and new cars of various model, design the system and give the Use case, Class and Object diagrams with the following requirements.
    - Customers are allowed to perform test drive before purchasing the car.
    - The company also sell car parts and do servicing
    - The customer either purchases the parts available in the part department directly or the service department can replace the parts of the car during servicing the car.
2. Consider a Car Company which sell both the used cars and new cars of various model, design the system and give the Object, Activity and State diagrams with the following requirements.
    - Customers are allowed to perform test drive before purchasing the car.
    - The company also sell car parts and do servicing
    - The customer either purchases the parts available in the part department directly or the service department can replace the parts of the car during servicing the car.
3. Consider a Car Company which sell both the used cars and new cars of various model, design the system and give the Interaction diagrams with the following requirements.
    - Customers are allowed to perform test drive before purchasing the car.
    - The company also sell car parts and do servicing

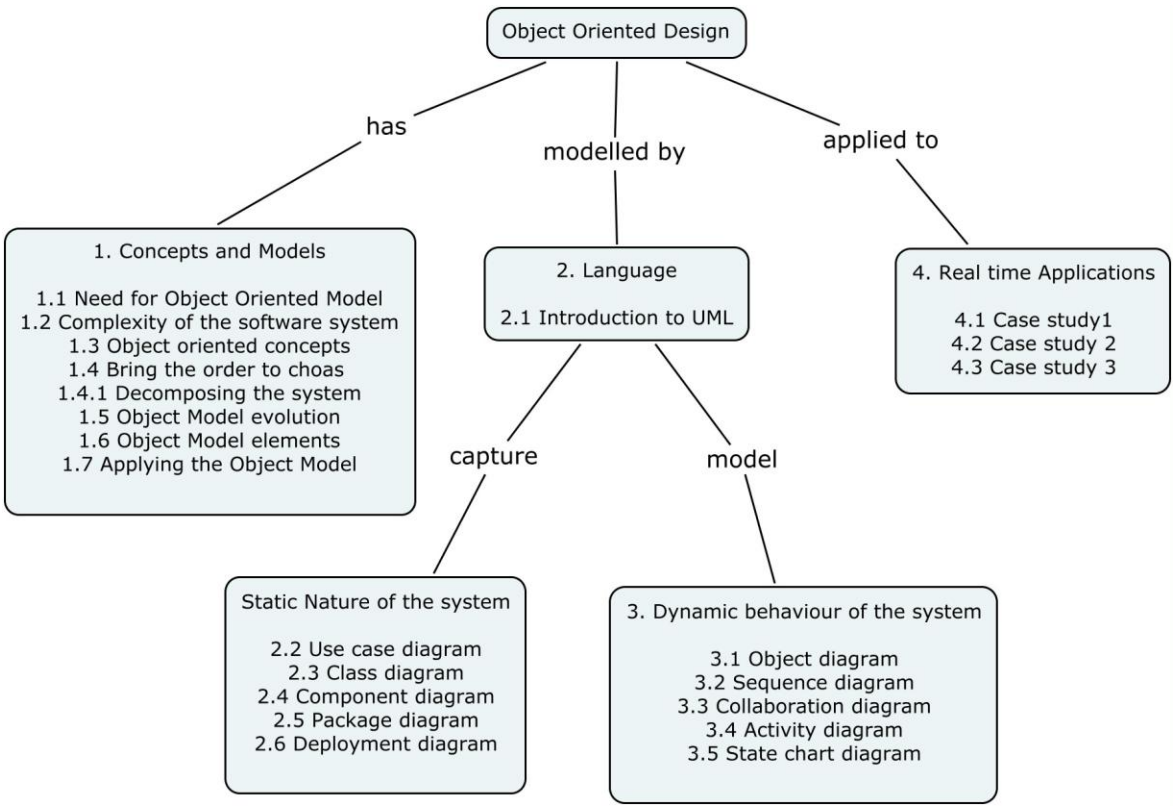
- The customer either purchases the parts available in the part department directly or the service department can replace the parts of the car during servicing the car.

4. For the Library Management System using UML draw the following diagrams.
   (i) Use Case (ii) Sequence (iii) Activity
5. Illustrate the following diagrams with suitable example.
   (i) Component (ii) Deployment (iii) Collaboration

Analyze

1. Analyze the Library Management System, overcome the complexity by decomposing the system and arrive the hierarchical structure for the system.
2. Compare and contrast the sequence and collaboration diagrams with suitable example.
3. Distinguish the static and dynamic behavior of the system.

Concept Map:



Syllabus:

Introduction – Need for Object Oriented Model – Complexity of the software system – Object oriented concepts - Bringing order to chaos – Decomposing the system - Evolution of Object model - Elements of object model - Applying the object model, **Unified Modeling Language : Static Diagram –** UML introduction – Use case diagram – Class diagram - Component diagram – Package and Deployment diagrams , **Unified**

Modeling Language : Dynamic Diagrams - Object diagram – Sequence diagrams **-** Collaboration diagrams **-** Activity diagrams - State chart diagram , **Applications of object Oriented Design –** Case studies

Text 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 Modeling Language – User Guide", Addison Wessley, 2nd Edition, 2005.

Reference Books:

1. Ali Bahrami, "Object Oriented System development ", McGraw Hill international edition 1999.
2. John Deacon, "Object Oriented Analysis and Design", Addison Wesley, First Edition 2005.

Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**INTRODUCTION (13)**	
1.1	Need for Object Oriented Model	2
1.2	Complexity of the system	2
1.3	Object oriented concepts	2
1.4	Bringing order to chaos	1
1.4.1	Decomposition of the system	2
1.5	Evolution of Object model	1
1.6	Elements of object model	2
1.7	Applying the object model	1
2	**UNIFIED MODELING LANGUAGE – STATIC DIAGRAMS (9)**	
2.1	UML introduction	1
2.2	Use case diagram	2

2.3	Class diagram	4
2.4	Component diagram	1
2.5	Package and Deployment diagrams	1
3	**UNIFIED MODELING LANGUAGE – DYNAMIC DIAGRAMS (14)**	
3.1	Object diagram	2
3.2	Sequence diagram	3
3.3	Collaboration diagram	3
3.4	Activity diagram	4
3.5	State chart diagram	2
4	**APPLICATION OF OOAD (9)**	
4.1	Case study 1	3
4.2	Case study 2	3
4.3	Case study 3	3
	Total	**45**

Course Designer:

1. Ms. A.M.Rajeswari – amrcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CGG	3	0	-	3

CGG Essentials of Mobile Application Development 3:0

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:

- Objected Oriented Programming Principles
- Networking Basics
- Basic knowledge on Java Programming

Program Outcomes addressed:

b. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.

d. Graduates will demonstrate an ability to design a system, component or process as per needs and specifications.

e. Graduates will demonstrate an ability to visualize and work on laboratory and multi-disciplinary tasks.

Competencies:

The Student will be able to

1. Describe the major mobile device platforms and their capabilities.
2. Apply the principles of web service architecture to mobile application development.
3. Explain the mobile platforms iOS and Android SDK.
4. Illustrate the Mobile Application architecture of Android.
5. Apply techniques of Android to leverage the design of an application.

Assessment Pattern:

	Bloom's Category	**Test 1**	**Test 2**	**Test3**	**End-semester examination**
1	Remember	40	30	20	20
2	Understand	20	30	20	20
3	Apply	40	40	30	40

4	Analyze	0	0	0	0
5	Evaluate	0	0	0	0
6	Create	0	0	30	20

Course Level Learning Objectives:

Remember

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.

Understand

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. Explain the device capabilities of iOS and Android based smart phones and Tablets.
5. Compare the different ways to design a mobile application.

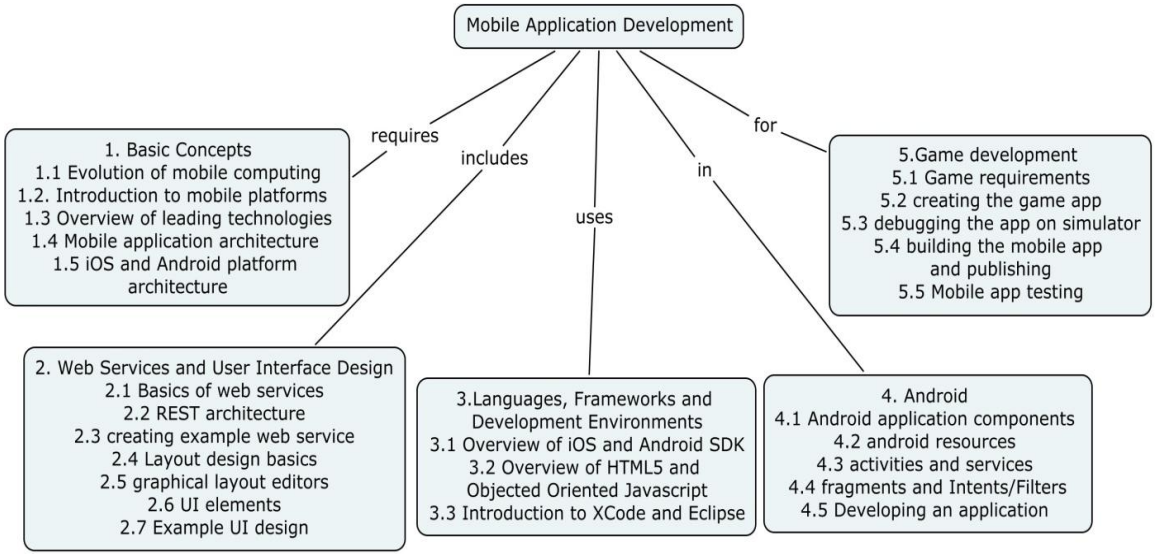
Apply

1. Develop a simple iOS notes application to take notes based on the text entered by the user and store the notes in a database.
2. Demonstrate the different ways to design a mobile application using Android SDK and HTML5.
3. Examine the online guides from Apple and Google on iOS and Android SDK.
4. Practice XCode IDE on Apple Mac Desktop and Eclipse IDE on Windows Desktop
5. Illustrate the steps to launch an mobile application on Android platform.

Create

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. 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.
4. Design a mobile web application with suitable UI Components.
5. 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.

References:

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:

No	Topic	No. of Lectures
1	**BASIC CONCEPTS (8)**	
1.1	Evolution of mobile computing	2
1.2	Introduction to mobile platforms	2
1.3	Overview of leading technologies available for mobile development	2
1.4	Mobile application architecture- iOS	1
1.5	Mobile application architecture- Android	1
2	**WEB SERVICES AND USER INTERFACE DESIGN(12)**	
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	2
2.6	UI elements	2
2.7	Example UI design	2
3	**LANGUAGES, FRAMEWORKS AND DEVELOPMENT ENVIRONMENTS(7)**	
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	2
3.5	Introduction to XCode and Eclipse development environments and device simulators	2
4	**ANDROID BASICS(11)**	
4.1	Android application components	2
4.2	Android resources	1
4.3	Activities and services	2

4.4	Fragments	2
4.5	Intents/Filters	2
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	**45**

Course Designers:

1. Ms. G. Madhu Priya (gmadhupriya@tce.edu)
2. Ms. M. Suguna (mscse@tce.edu)

Sub Code	Lectures	Tutorial	Practical	Credit
CGH	3	0	-	3

CGH Animation: Theory and Practice 3:0

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.

Programme Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.

f. Graduate will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

Competencies

1. Explain the role of animation in human computer interface.
2. Explain the principles and applications of 2D animation.
3. Explain and compare algorithms for visible surface detection algorithms.
4. Explain the role of modeling and rendering in 3D.
5. Select 2D and 3D tools for different applications.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3	End-semester examination
1	Remember	30	10	10	10
2	Understand	30	30	20	20
3	Apply	30	40	50	50
4	Analyze	10	20	20	20
5	Evaluate	0	0	0	0
6	Create	0	0	0	0

Course Level Learning Objectives

Remember

1. Name the basic 2D transformations.
2. Give the matrix representation for 2D and 3D Scaling.
3. Define projection and its types.
4. Define animation.
5. Mention some general purpose languages that support animation.
6. What is scientific visualization?

7. List out the visualization techniques.
8. What are the display methods available for representing 3D objects?
9. List out 2D and 3D tools.
10. What are the steps involved in designing animation sequence.
11. Define rendering.

Understand

1. Difference between raster and vector graphics.
2. What is the difference between 2D and 3D objects?
3. Explain the techniques of visualization with an example.
4. How will you perform morphing? Mention its applications.
5. Explain how to create running tiger from moving automobile using morphing effect.
6. Write a procedure to create bouncing ball using motion specifications.
7. Differentiate parallel projection from perspective projection.
8. Explain the applications of 3D animation in detail.

Apply

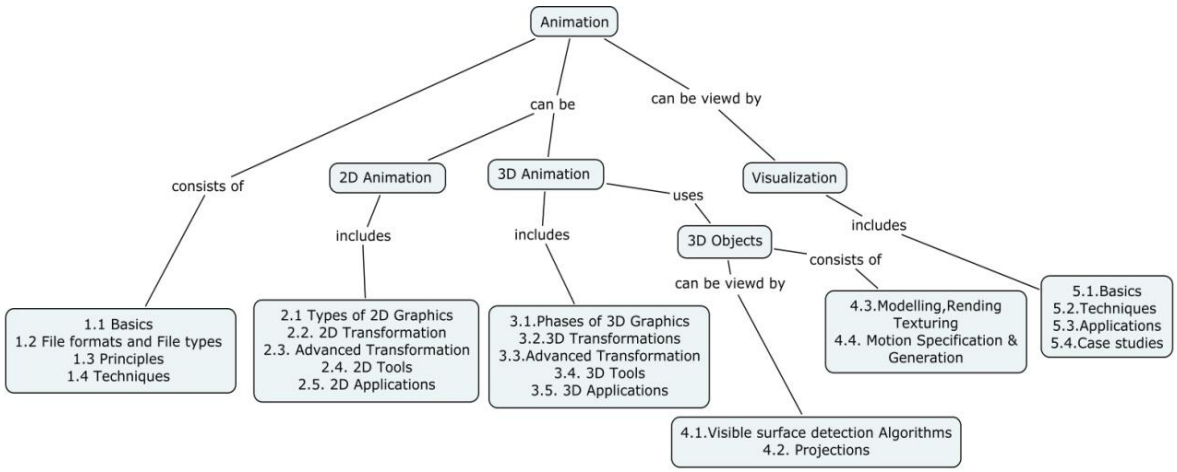
1. Illustrate the use of scan-line algorithm for displaying the visible-surfaces of a polyhedron.
2. Illustrate the use of depth buffer method for displaying the visible-surfaces of 3D objects.
3. Exhibit how the visualization techniques can be used in real time scenario?
4. Reflect a diamond shaped polygon whose vertices are A (-1,0), B(0,-2), C(1,0) and D(0,2) about
    a) about the horizontal line y=3
    b) about the vertical line x=4
    c) about the line y=x+3
5. Illustrate the effect of shearing transformations on the square A(0,0), B(2,0), C(2,2) and d(0,2)
    a) X direction sheariong when shx=-3
    b) Y direction shearing with respect to xref =3 and shy=2
    c) Shearing in both directions when shx=4 and shy=5
6. Use motion specifications to create a bouncing ball.

Analyze

1. Analyze the space and time complexity of 2D Graphics.
2. Analyze the computational efficiency in 2D composite transformations.
3. Compare and contrast three methods of visible surface detection algorithm and hence analyze which algorithm is efficient?

4. Analyze how parallel projection is differed from perspective projection .Explain different types of parallel projection and hence obtain the transformation matrix for the same.
5. Explain different types of perspective projection and hence obtain the transformation matrix for the same.
6. Explain the applications of visualization in different fields.

Concept Map



Syllabus

INTRODUCTION TO ANIMATION: Animation basics, File formats and File types, Principles of animation, Techniques of animation – Traditional animation, stop motion and computer animation, Introduction to 2D and 3D animation. **2D ANIMATION-** Types of 2D Graphics, 2D Transformations- Translation, Rotation, Scaling, 2D Advanced Transformations- Reflection and Shearing, 2D Animation Tools Introduction, Applications of 2D animation. **3D ANIMATION -** Phases of 3D graphics, 3D Transformations- Translation, Rotation, Scaling, 3D Advanced Transformations -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 Specifications and Generation. **VISUALIZATION -** Visualization- Basics, Techniques of Visualization, Applications of Visualization, Case Studies.

References:

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 and File types	1
1.3	Principles of animation	1
1.4	Techniques of animation – Traditional animation, stop motion and computer	3
1.5	Introduction to 2D and 3D animation	1
2	**2D ANIMATION**	
2.1	Types of 2D Graphics	1
2.2	2D Transformations - Translation, Rotation, Scaling	3
2.3	2D Advanced Transformations - Reflection and Shearing	3
2.4	2D Animation Tools Introduction	4
2.5	Applications of 2D animation	1
3.	**3D ANIMATION**	
3.1	Phases of 3D Graphics	1
3.2	3D Transformations- Translation, Rotation, Scaling	3
3.3	3D Advanced Transformations - Reflection and Shearing	3

3.4	3D Animation Tools Introduction	5
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	3
4.3	3D Requirements- Modeling, Rendering, Texturing	5
4.4	Motion Specifications and Generation	2
5	**VISUALIZATION**	
5.1	Visualization- Basics	1
5.2	Techniques of Visualization	1
5.3	Applications of Visualization	1
5.4	Case Studies	2
	Total	**50**

Course Designers:

1. Ms. S. Sridevi sridevi@tce.edu
2. Ms. R. Suganya rsuganya@tce.edu