Annexure – K

CURRICULUM AND DETAILED SYLLABI

FOR

M.E DEGREE (Computer Science and Engineering) PROGRAMME

FIRST SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2011-2012 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

Department of Computer Science and Engineering

Graduating Students of ME 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 the M.E. CSE (for those who are admitted from the year 2011-2012)

4th (12)							Project Phase II 0:12
3rd (16)	CT31 Data Warehousing and Mining Techniques 4:0	CTX Elective 5 4:0	CTX Elective 6 4:0				Project Phase I 0:4
2nd (24)	CT21 Modeling and Simulation 3:1	CT22 Software Engineering Theory and Practice 3:0	CTX Elective 1 4:0	CTX Elective 2 4:0	CTX Elective 3 4:0	CTX Elective 4 4:0	Seminar 0:1
1st (24)	CT11 Computer Architecture 4:0	CT12 Data Structures and Algorithms 3:1	CT13 Object Oriented Analysis and Design 3: 0	CT14 Network Technology 3:1	CT15 Compiler Design 3:1	CT16 Graphs and Combinatorial Algorithms 3:1	CT17 Analysis and Design Lab 0:1

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

(An Autonomous Institution affiliated to Anna University)

CURRICULAM

(For the Students admitted from the academic year 2011-2012)

Name of the Degree: M.E (Computer Science and Engineering) Programme

SUBJECTS OF STUDY

I SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT11	Computer Architecture	4	0	0	4
CT12	Data Structures and Algorithms	3	1	0	4
CT13	Object Oriented Analysis and Design	3	0	0	3
CT14	Network Technology	3	1	0	4
CT15	Compiler Design	3	1	0	4
CT16	Graphs and Combinatorial Algorithms	3	1	0	4
Practical:					
CT17	Analysis and Design Laboratory	0	0	3	1
		Total Credits			**24**

II SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT21	Modeling and Simulation	3	1	0	4
CT22	Software Engineering Theory and Practice	3	0	0	3
CTX	Elective 1	4	0	0	4
CTX	Elective 2	4	0	0	4
CTX	Elective 3	4	0	0	4
CTX	Elective 4	4	0	0	4
Practical:					
CT27	Seminar	0	0	3	1
		Total Credits			**24**

III SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT31	Data Warehousing and Mining Techniques	4	0	0	4
CTX	Elective 5	4	0	0	4
CTX	Elective 6	4	0	0	4
Practical:					
CT34	Project	0	0	12	4
		Total Credits			**16**

IV SEMESTER

Practical:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT41	Project	0	0	36	12
		Total Credits			**12**

Total No. of credits to be earned for the award of degree: 76

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E Degree (Computer Science and Engineering) Programme
SCHEME OF EXAMINATIONS
(For the candidates admitted from 2011-2012 onwards)

I 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	CT11	Computer Architecture	3	50	50	100	25	50
2	CT12	Data Structures and Algorithms	3	50	50	100	25	50
3	CT13	Object Oriented Analysis and Design	3	50	50	100	25	50
4	CT14	Network Technology	3	50	50	100	25	50
5	CT15	Compiler Design	3	50	50	100	25	50
6	CT16	Graphs and Combinatorial Algorithms						
PRACTICAL								
7	CT17	Analysis and Design Laboratory	3	50	50	100	25	50

II 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	CT21	Modeling and Simulation	3	50	50	100	25	50
2	CT22	Software Engineering Theory and	3	50	50	100	25	50

		Practice						
3	CTX	Elective 1	3	50	50	100	25	50
4	CTX	Elective 2	3	50	50	100	25	50
5	CTX	Elective 3	3	50	50	100	25	50
6	CTX	Elective 4	3	50	50	100	25	50
PRACTICAL								
7	CT27	Seminar	3	50	50	100	25	50

III 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	CT31	Data Warehousing and Mining Techniques	3	50	50	100	25	50
2	CTX	Elective 5	3	50	50	100	25	50
3	CTX	Elective 6	3	50	50	100	25	50
PRACTICAL								
4	CT34	Project	-	150	150	300	75	150

IV 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	CT41	Project	-	150	150	300	75	150

Sub Code	Lectures	Tutorial	Practical	Credit
C T11	4	0	-	4

CT11 Computer Architecture 4:0

Preamble: The computer architecture has evolved—from a world of mainframes, minicomputers, and microprocessors, to a world dominated by microprocessors, and now into a world where microprocessors themselves are encompassing all the complexity of mainframe computer. The computer architect's role is to combine a thorough understanding of the state of the art of what is possible, a thorough understanding of the historical and current styles of what is desirable, a sense of design to conceive a harmonious total system, and the confidence and energy to marshal this knowledge and available resources to go out and get something built. To accomplish this, the architect needs a tremendous density of information with an in-depth understanding of the fundamentals. This course introduces the taxonomies of computer design and the basic concerns of computer architecture, gives an overview of the technology trends that drive the industry, and lays out the methods to using all this information in the art of computer design.

Competencies

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

1. Explain what Computer performance measures are.
2. Describe the issues involved in Instruction set design
3. Exploit ILP using dynamic scheduling method.
4. Determine the optimization methods for reducing the cache miss penalty.
5. Determine the state transitions for a given sequence of memory 'reads' and 'writes' using Write invalidation cache coherence protocol.
6. Determine if in a pipelined system there exists a structural hazard or not.

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. Define Principle of locality.
2. What do you mean by antidependance ?
3. What is cache coherence?
4. Define Thread Level parallelism.
5. What are structural hazards?
6. What is the purpose of the reorder buffer?

Understand

1. What are the choices on encoding an Instruction Set Architecture?
2. What the possible hazards which are to be taken care of to exploit ILP.
3. List the limitations of symmetric shared memory multiprocessors.
4. Which cache optimization improves temporal locality to reduce misses.
5. Differentiate synchronous vs. Asynchronous exceptions
6. Give the significance of the branch delay slot.

Apply

1. Your company has just bought a new dual Pentium processor, and you have been tasked with optimizing your software for this processor. You will run two applications on this dual Pentium, but the resource requirements are not equal. The first application needs 80% of the resources, and the other only 20% of the resources. Given that 40% of the first application is parallelizable, how much speedup would you achieve with that application if run in isolation?
2. Show what the information tables look like for the following code sequence when only the first load has completed and written its result:
   L.D F6,32(R2)
   L.D F2,44(R3)
   MUL.D F0,F2,F4
   SUB.D F8,F2,F6
   DIV.D F10,F0,F6
   ADD.D F6,F8,F2
3. Assume that words x1 and x2 are in the same cache block, which is in the shared state in the caches of both P1 and P2. Assuming the following sequence of events, identify each miss as a true sharing miss, a false sharing miss, or a hit. Any miss that would occur if the block size were one word is designated a true sharing miss.

Time	P1	P2
1	Write x1	
2		Read x2
3	Write x1	
4		Write x2
5	Read x2	

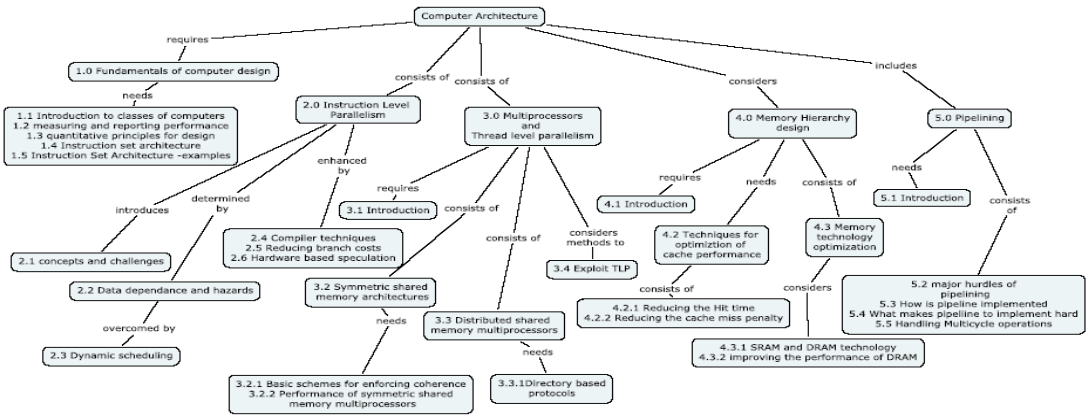
4. For the code below, determine which accesses are likely to cause data cache misses. Next, insert prefetch instructions to reduce misses. Finally, calculate the number of prefetch instructions executed and the misses avoided by prefetching. Let's assume we have an 8 KB direct-mapped data cache with 16-byte blocks, and it is a write-back cache that does write allocate. The elements of a and b are 8 bytes long since they are double-precision floating-point arrays. There are 3 Rows and 100 columns for a and 101 rows and 3 columns for b. Let's also assume They are not in the cache at the start of the program.

```
for (i = 0; i < 3; i = i+1)
for (j = 0; j < 100; j = j+1)
a[i][j] = b[j][0] * b[j+1][0];
```

4. Give the actions to be taken in the pipeline control to save the pipeline state, when an exception occurs.

6. Consider the unpipelined processor in the previous section. Assume that it has a 1 ns clock cycle and that it uses 4 cycles for ALU operations and branches and 5 cycles for memory operations. Assume that the relative frequencies of these operations are 40%, 20%, and 40%, respectively. Suppose that due to clock skew and setup, pipelining the processor adds 0.2 ns of overhead to the clock. Ignoring any latency impact, how much speedup in the instruction execution rate will we gain from a pipeline?

Concept Map:



Syllabus

Fundamentals of Computer Design-Introduction to classes of computers, Defining computer architecture- PowerPC architecture.-Measuring and reporting performance-Quantitative principles of computer design-Instruction set Architecture- classification. Instruction Set Architecture – examples. **Instruction Level Parallelism-**Concepts and challenges, Data dependence and hazards- Basic Compiler techniques for exposing ILP-Reducing branch costs with predictions-Dynamic scheduling- Hardware based speculation. **Multiprocessors and Thread level parallelism-**Introduction-Symmetric shared memory architectures-Basic schemes for enforcing coherence-Performance of symmetric shared memory multiprocessors-Distributed shared memory multiprocessors-Directory based protocols-Exploiting Thread level parallelism-future trends of processors. **Memory hierarchy design-**Introduction-Techniques for optimization of cache performance-Reducing the Hit time-Reducing cache miss penalty-Memory technology optimization-SRAM and DRAM technology-Improving the performance of DRAM. **Pipelining-** Introduction-Major hurdles of pipelining-How is pipeline implemented-What makes pipeline to implement hard-Handling multicycle operations.

References:

1. John L Hennessy and David A. Patterson, Computer Architecture- A Quantitative Approach, Morgan Kauffman publishers, Fourth Edition, 2007
2. William stallings , Computer Organization and architecture, Prentice Hall, Fifth Edition, 2002
3. Kai Hwang, Advanced Computer Architecture, Tata McGrawhill, Eighteenth Reprint, 2008.
4. John Paul shen,Mikko H Lipasti, Modern Processor design, Tata McGrawhill, 2005.

Course contents and Lecture Schedule:

	Lecture Topics	**Lecture Periods**
1.0	**Fundamentals of Computer Design**	
1.1	Introduction to classes of computers, Defining computer architecture, PowerPC architecture	2
1.2	Measuring and reporting performance	2
1.3	Quantitative principles of computer design	2
1.4	Instruction set Architecture- classification	2
1.5	Instruction Set Architecture – examples	1
2.0	**Instruction Level Parallelism**	
2.1	Concepts and challenges	1
2.2	Data dependence and hazards	2

2.3	Dynamic scheduling	3
2.4	Basic Compiler techniques for exposing ILP	2
2.5	Reducing branch costs with predictions	2
2.6	Hardware based speculation	2
3.0	**Multiprocessors and Thread level parallelism**	
3.1	Introduction	1
3.2	Symmetric shared memory architectures	1
3.2.1	Basic schemes for enforcing coherence	2
3.2.2	Performance of symmetric shared memory multiprocessors	1
3.3	Distributed shared memory multiprocessors	2
3.3.1	Directory based protocols	2
3.4	Exploiting Thread level parallelism –future trend of processors	2
4.0	**Memory hierarchy design**	
4.1	Introduction	1
4.2	Techniques for optimization of cache performance	1
4.2.1	Reducing the Hit time	1
4.2.2	Reducing cache miss penalty	2
4.3	Memory technology optimization	1
4.3.1	SRAM and DRAM technology	1
4.3.2	Improving the performance of DRAM	2
5.0	**Pipelining**	
5.1	Introduction	1
5.2	Major hurdles of pipelining	2
5.3	How is pipeline implemented	2
5.4	What makes pipeline to implement hard	2
5.5	Handling multicycle operations	2
	Total number of Hours	50

Course Designer:

P.Chitra pccse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT12	3	1	-	4

Ct12 Data Structures and Algorithms 3:1

Preamble: This course will facilitate the Students to identify, formulate and solve real world engineering problems that require usage of algorithms. Students learn different algorithmic techniques like greedy algorithm, dynamic programming, divide and conquer as part of tree and graph algorithms. They use these techniques to solve problems both in theoretical design and practical implementations.

Competencies

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

1. Understand the role of algorithm in technology and estimate the growth of functions.
2. 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)).
3. Determine the time complexity of algorithms for sorting (insertion, selection, merge, quick sort and heap sort), dynamic programming, searching (binary search tree and red black tree) and graphs (directed and undirected) applied to average, worst and best cases.
4. Apply algorithms for shortest path, matrix multiplication, minimum spanning tree, and traveling salesman problems
5. Apply algorithms for NP hard and NP complete problem.
6. Analyze the Complexity of the problem and prove it by reduction techniques.
7. Analyze whether a problem like traveling salesperson and vertex-cover problem has an algorithmic solution.

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	40
4	Analyze	20	30	40
5	Evaluate	0	0	0
6	Create	0	0	0

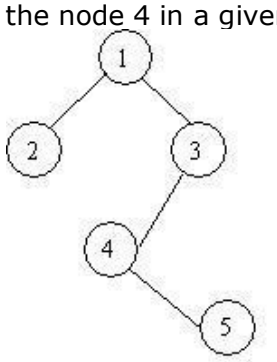
Course Level Learning Objectives:

Remember:

1. Define: Asymptotic notations
2. List out the steps involved in solving recurrence equations
3. What is best and worst case analysis?
4. What is meant by minimum spanning tree?
5. What is the use of hashing?
6. What are the properties of B tree?

Understand:

1. At what location can you store the node 4 in a given binary tree using array?



2. Sort the given values using Quick Sort?

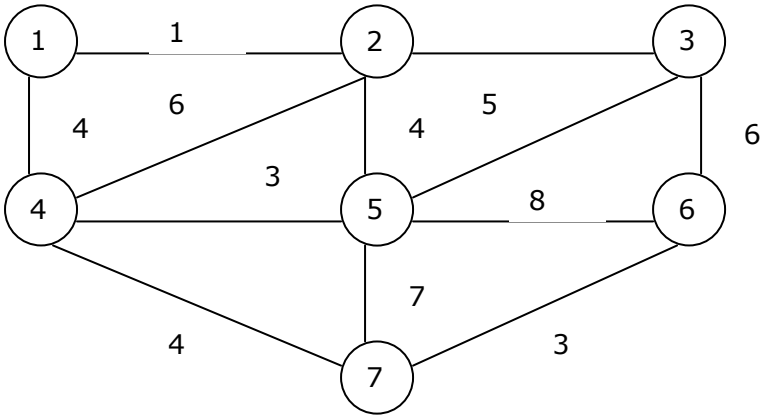
65	70	75	80	85	60	55	50	45

3. Classify the Hashing Functions based on the methods by which the key value is found.
4. Differentiate Dynamic programming from Greedy algorithm.
5. Mention the two differences between open and closed hashing technique.
6. Differentiate P and NP classes
7. How does AVL tree differ from Binary search tree?

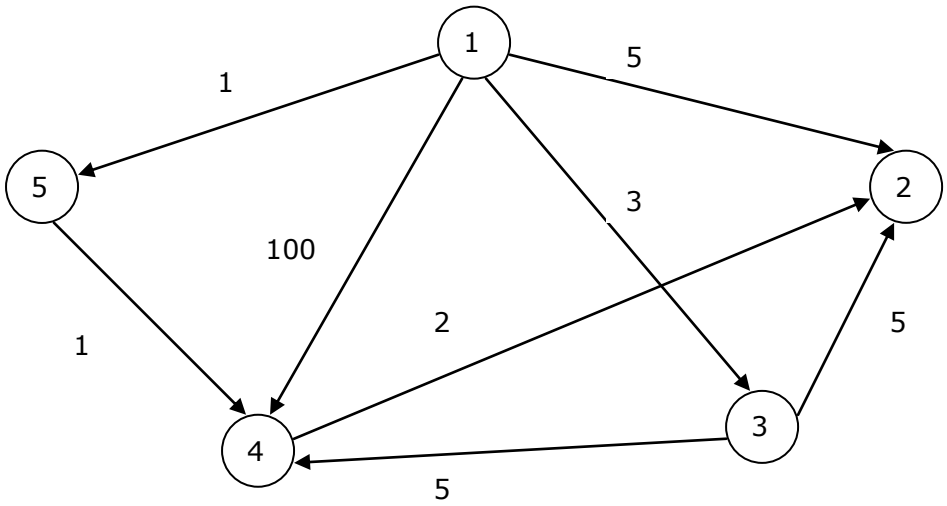
Apply:

1. 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
2. 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).
3. 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.)

4. Apply Heap sort on the array A= {5, 13, 2, 23, 7, 17, 20, 8, 11}?pply the Kruskal's algorithm for following minimu 2 ıning tree.



5. Apply the **dijikstra** algorithm for the graph



6. Calculate the product ABCD of four matrixes applying chained matrix multiplication

   A is 13 * 5 C is 89 * 3

   B is 5 * 89 D is 3 * 34

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?

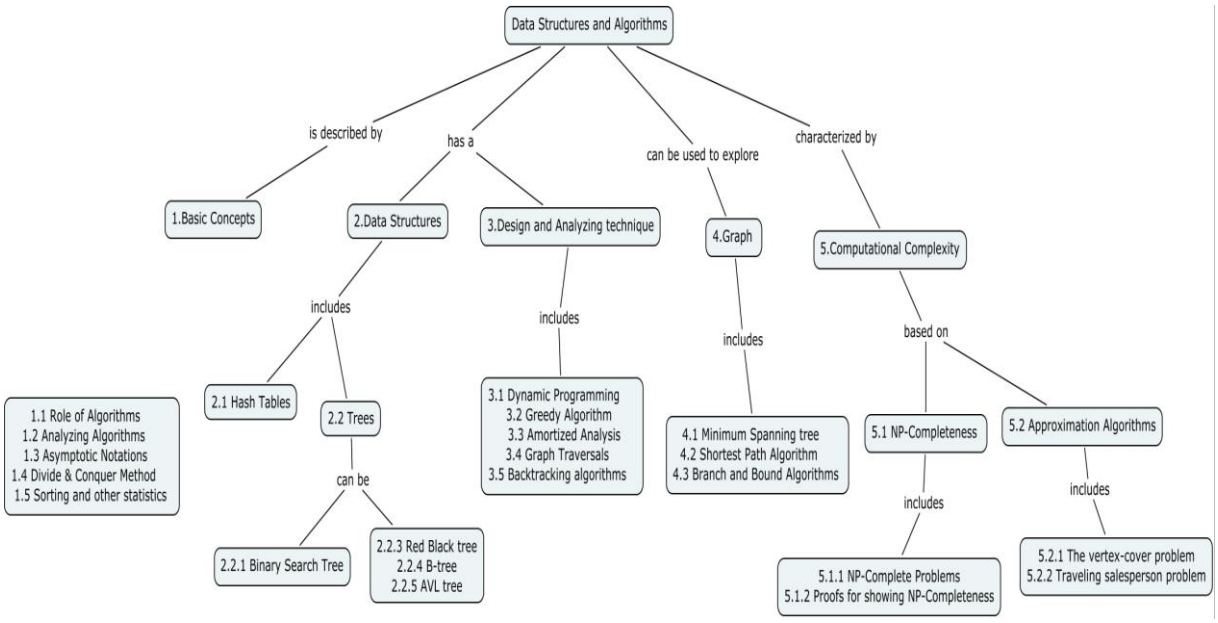
   A. O(log n)
   B. O(n)
   C. O(n log n)
   D. O(n^2)

4. Analyze the Travleing salesperson problem and show their NP-Completeness

5. Analyze the Heap sort based on worst, best and average case analysis.

6. Analyze the complexity of approximation algorithm for the vertex-cover problem

Concept Map:



Syllabus

Review of Basic concepts: The role algorithms in computing, Analyzing algorithms, Growth of functions, Basic concepts in Divide and Conquer, Sorting and order statistics- Heap Sort and Quick sort. **Data Structures:** Hash Tables, Binary Search Trees, Red Black Trees, B-Trees, AVL Tree. **Advanced Design and Analysis Technique:** Dynamic Programming, Greedy Algorithms, Amortized Analysis, Exploring Graphs, Backtracking. **Graph algorithms:** Minimum Spanning trees, Shortest Path Algorithms, Branch and Bound Techniques. **Computational Complexity:** NP- Completeness, and Approximation algorithms

References :

1.Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "I ntroduction to Algorithms", PHI Learning Private Limited, Third Edition, 2009.

2.Sartaj Sahni, " Data Structures, Algorithms and Applications in C++", Tata McGraw Hill, 2000.

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

4.Gilles Brassard and Paul Bratley, "Fundamentals of Algorithms", Prentice Hall of India, 2007.

Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Basics (9)**	
1.1	The role of algorithms in computing	1
1.2	Analyzing algorithms- Best, Worst and Average case	2
1.3	Growth of functions- Asymptotic Notations - Tutorial on Asymptotic Complexities	1
1.4	Divide and Conquer concepts for solving Recurrences	2
1.5	Sorting and order statistics-Heap Sort – Tutorial	2
1.6	Quick sort	1
2.	**Data Structures (8)**	
2.1	Hash Tables	1
2.2	Trees- Unbalanced and Unbalanced	1
2.2.1	Binary Search Trees – Tutorial	1
2.2.2	Red Black Trees	2
2.2.3	B-Trees – Tutorial on deletion	1
2.2.4	AVL Tree	2
3.	**Advanced Design and Analysis Techniques (11)**	
3.1	Dynamic Programming- Matrix-chain Multiplication	1
3.1.1	Elements of Dynamic Programming	1
3.2	Greedy Algorithm - Elements of Greedy strategy	1

3.2.1	Huffman Codes – Tutorial	2
3.3	Amortized Analysis	2
3.4	Exploring Graphs and Traversals – Tutorial	2
3.5	Backtracking-Traveling Salesperson, The Queen's Problem	2
4	**Graph Algorithms (6)**	
4.1	Minimum Spanning Trees- Kruskal's and Prim's Algorithm	1
4.2	Single Source Shortest path algorithm – Dijkstra' s Algorithm, All Pair's Shortest Path Algorithm – Tutorial	1
4.3	Branch and Bound Algorithms	1
4.3.1	Traveling Salesperson Problem – Tutorial	1
4.3.2	Job Scheduling Problem	2
5	**Computational Complexity (6)**	
5.1	NP - Completeness	1
5.1.1	NP Complete Problems	1
5.1.2	NP-Completeness and Reducibility – Tutorial	1
5.2	Approximation Algorithms	1
5.2.1	The vertex - cover problem	1
5.2.2	The Traveling salesperson problem – Tutorial	1
	Total Periods	**40**

Course Designer

1. Dr.S.Padmavathi spmcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT13	3	0	-	3

CT13 Object Oriented Analysis and Design 3:0

Preamble: This Objective of this subject is to promote the practice of object oriented analysis and design concepts at a higher level of abstraction. This subject covers the object oriented techniques, representations and patterns used to analyze and model a system and it also deals with all important Unified Modeling Language(UML) elements for good object oriented analysis and design

Competencies

1. Explain the basics of object oriented programming.
2. Discuss a graphical modeling language called UML to take control of the software development.
3. Explain the analysis of the software to be developed using an Object Oriented approach.
4. Discuss the design of the software to be developed considering the Object type model.
5. Explain the design of the software to be developed using the technical model.

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. Recall at least two definitions of objects and of object-orientation.
2. What is the difference between a model and a prototype?
3. What is the difference between analysis and design
4. Differentiate between stamp and data coupling.
5. What is aggregation?
6. What are the possible candidates of use cases that turn into classes and methods?
7. What is the technique used to depict common behavior?
8. What is coupling and cohesion?
9. Who is an actor?
10. What is system analysis?

11. State the need for information hiding.
12. What is the use of state machines.

Understand

7. Describe how, in the face of changing problems, practices and people, analysis has evolved, or should have evolved.
8. Consider one or two subject matters that you have a degree of familiarity with. Note down a few of the key creatures or entities.
9. Rewrite the following requirements to be clearer and more precise. They don't necessarily have to be correct. In other words, your job is to make them less ambiguous, you won't necessarily have the information to make them more correct.

    "The system must accept crimes."

    "Match fingerprints."

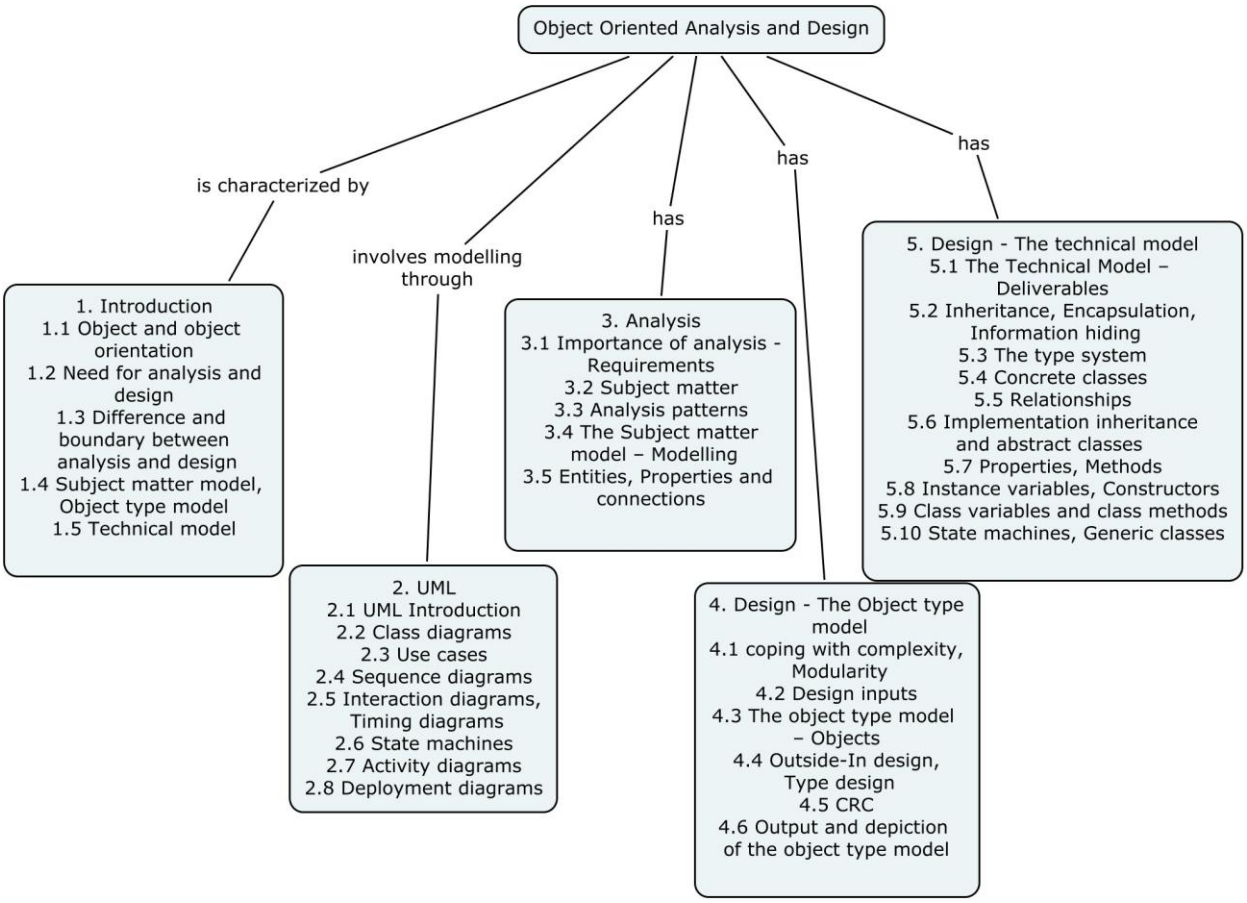
    "Produce bills."
10. Write up two important and illustrative use case scenarios for any hotel system development.
11. Consider a music player such as a CD, or MP3 player. Consider the random or shuffle play function that they often provide. As an exercise in abstraction, explain the meaning of shuffle, in words.
12. If you work in, or know something about, a subject matter like 'weather reporting and forecasting, list the entities you think might be relevant to a typical piece of software that concerned itself with that subject matter?
13. Briefly describe all the reasons why we might need to be more careful with generalization relationships than with associations and aggregations

Apply

1. Decide on the packages you would use in order to organize the development of a new web browser? What influenced you in your choice of packages?
2. Discuss the generalization relationships that one might find and model in subject matters ("the real world") with the inheritance mechanism in the programming language you know best.
3. Draw the following UML diagrams for a Digital Library System.
    i) Use case diagram
    ii) Activity diagram
    iii) Sequence diagram
    iv) State chart diagram

4. Consider the design of a library management system and illustrate on the ways to achieve a highly cohesive design.
5. Draw use case diagram to show what a car maintenance system is, without getting into unnecessary detail.
6. Show a design where it is best to use delegation rather than inheritance.

Concept Map:



Syllabus:

Introduction – Object and object orientation –Need for analysis and design – Difference and boundary between analysis and design – **Three models –** Subject matter model – Object type model – Technical model – UML introduction – Class diagrams **-** Use cases **-** Sequence diagrams **-** Interaction diagrams **-** Timing diagrams **-** State machines **-** Activity diagrams **-** Deployment diagrams **Analysis** – Importance of analysis – Requirements – Subject matter – Analysis patterns – **The Subject matter model –** Modelling – Entities – Properties and connections –**Design –** coping with complexity – Modularity –Design inputs - **The object type model –** Objects – Outside-In design – Type design – CRC - Output and depiction of the object type model –**The Technical Model** – Deliverables –

Inheritance – Encapsulation – Information hiding – The type system – Concrete classes – Relationships – Implementation inheritance and abstract classes – Properties – Methods– Instance variables – Constructors – Class variables and class methods – State machines – Generic classes

References:

1. John Deacon, Object Oriented Analysis and Design, First Edition, Addison Wesley, 2005.

2. Brett McLaughlin, Gary Pollice, David West, Head First Object-Oriented Analysis and Design, O'Reilly Media, 2006.

3.Grady Booch, Robert A.Maksimchuk, Michael W.Engel, Bobbi J.Young, Jim Conallen, Kelli A. Houston, Object-Oriented Analysis and Design with Applications, Third Edition, Addison-Wesley, 2011.

Course contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Introduction(5)**	
1.1	Object and object orientation	1
1.2	Need for analysis and design	1
1.3	Difference and boundary between analysis and design	1
1.4	Subject matter model, Object type model	1
1.5	Technical model	1
2	**UML(11)**	
2.1	UML introduction	1
2.2	Class diagrams	2
2.3	Use cases	1
2.4	Sequence diagrams	2
2.5	Interaction diagrams, Timing diagrams	1
2.6	State machines	2
2.7	Activity diagrams	1
2.8	Deployment diagrams	1
3	**Analysis(6)**	

3.1	Importance of analysis -Requirements	1
3.2	Subject matter	1
3.3	Analysis patterns	1
3.4	The Subject matter model **–** Modelling	1
3.5	Entities, Properties and connections	2
4	**Design – The object type model(7)**	
4.1	coping with complexity, Modularity	1
4.2	Design inputs	1
4.3	**The object type model –** Objects	1
4.4	Outside-In design, Type design	1
4.5	CRC	2
4.6	Output and depiction of the object type model	1
5	**Design – The Technical model(11)**	
5.1	**The Technical Model** – Deliverables	1
5.2	Inheritance, Encapsulation, Information hiding	1
5.3	The type system	1
5.4	Concrete classes	1
5.5	Relationships	1
5.6	Implementation inheritance and abstract classes	2
5.7	Properties, Methods	1
5.8	Instance variables, Constructors	1
5.9	Class variables and class methods	1
5.10	State machines, Generic classes	1
	Total No of Hours	40

Course Designer:

J. Jane Rubel Angelina janerubel@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT14	3	1	-	4

CT14 Network Technology 3:1

Preamble:

Network Technology is needed to make sure the computers are interconnected properly and the communication between the systems are done effectively so that the underlying process is understandable and data is transferred in a secured manner.

Competencies

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

1. Explain the basic data communication model
2. Detect and correct error in data that may occur during transmission
3. Differentiate different protocols used in the network and transport layers
4. Select a specific routing algorithm based on the application need
5. Identify methods to minimize congestion in the network
6. Identify issues for maintaining Quality of Service

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	30
4	Analyze	0	0	20
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives:

Remember

1. Draw the OSI reference model?
2. Describe sliding window protocol?
3. What is meant by CSMA/CD and CSMA/CA?
4. What is DNS?
5. What is QOS?
6. Why the protocol field in Ipv4 header in not present in IPv6 header?
7. Define UDP?

Understand

1. Compare OSI and TCP Reference models?

2. Explain the performance Issues in Ethernet?
3. Compare the different Ethernet types?
4. Describe the Frame Structure of MAC Sub layer?
5. Explain the method to reassemble IP fragments at the destination?
6. Describe the major difference between warning bit and the RED method?
7. Summarize the various services to maintain QOS?

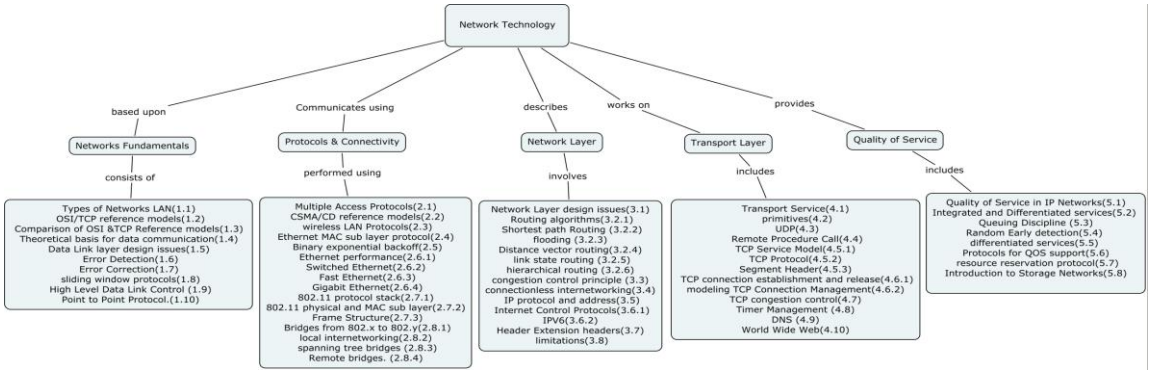
Apply

1. How to detect error in data link layer?
2. Classify the different Routing algorithms used?
3. Show how a bit String '0111101111101111110' is being transmitted at the data link layer after performing bit stuffing?
4. Ten thousand airline reservation stations are competing for the use of a single slotted ALOHA channel. The average station makes 18 request/hour. A slot is 125 micro second. Calculate the approximate channel load?
5. Suppose that the TCP congestion window is set to 18KB and a timeout occurs. How big will the window be if the next 4 transmission bursts are all successful assuming that the maximum segment size is 1KB?

Analyze

1. Find out the reminder obtained by dividing X^7+X^5+1 by the generator polynomial X^3+1?
2. Compare the various Collusion free protocols?
3. Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is low. Justify your answer?
4. A network on the Internet has a subnet mask of 255.255.240.0.predict the maximum number of hosts it can handle?

Concept Map



Syllabus

Network Fundamentals: Types of Networks, LAN, OSI reference models, TCP reference models, Comparison between OSI &TCP Reference models, Theoretical basis for data

communication, Data Link layer design issues, Error Detection, Error Correction, sliding window protocols, High Level Data Link Control & Point to Point Protocol. **Protocols and Connectivity**: Multiple Access Protocols, CSMA/CD reference models, wireless LAN Protocols, Ethernet MAC sub layer protocol, Binary exponential back off, Ethernet performance, Switched Ethernet, Fast Ethernet, Gigabit Ethernet,*02.11 protocol stack,802.11 physical and MAC sub layer ,Frame Structure, Bridges from 802.x to 802.y,local internetworking, spanning tree bridges ,Remote bridges. **Network Layer:** Network Layer design issues, Routing algorithms, Shortest path Routing ,flooding ,Distance vector routing, link state routing , hierarchical routing, congestion control principle, connectionless internetworking, IP protocol and address, Internet Control Protocols,IPV6,Header Extension headers, limitations. **Transport Layer and DNS:** Transport Service, primitives, UDP, Remote Procedure Call, TCP Service Model ,TCP Protocol, Segment Header, TCP connection establishment and release, modelling TCP Connection Management ,TCP congestion control, Timer Management ,DNS , World Wide Web. **Quality of Services:** Quality of Service in IP Networks, Integrated and Differentiated services, Queuing Discipline, Random Early detection, differentiated services, Protocols for QOS support, resource reservation protocol, Introduction to Storage Networks, recent advancements in networks

References:

1. Andrew S. Tanenbaum, Computer Networks, 4th Edition, Pearson Education , 2003
2. William Stallings, High Speed Networks and Internet, 2nd Edition Pearson Education, 2003.
3. William Stallings, Data and Computer Communication, 7th Edition Pearson Education , 2003

Course Content and Lecture Schedule

No	Topic	No. of Lectures
1	**Network Fundamentals**	**(8)**
1.1	Types of Networks ,LAN ,WAN	1
1.2	OSI /TCP reference models and comparison	1
1.3	Theoretical basis for data communication	1
1.4	Data Link layer design issues	1
1.5	Error Detection and Error Correction-Tutorial	1

1.6	sliding window protocols	1
1.7	High Level Data Link Control &	1
1.8	Point to Point Protocol	1
2	**Protocols and Connectivity**	**(7)**
2.1	Multiple Access Protocols	1
2.2	CSMA/CD reference Models	1
2.3	wireless LAN Protocols and Ethernet MAC sub layer protocol	1
2.4	Binary exponential back off	1
2.5	Ethernet performance Switched Ethernet Fast Ethernet Gigabit Ethernet	1
2.6	802.11 protocol stack,802.11 physical and MAC sub layer, Frame Structure	1
2.7	Bridges from 802.x to 802.y,local internetworking spanning tree bridges Remote bridges – **Tutorial**	1
3	**Network Layer:**	**(9)**
3.1	Network Layer design issues	1
3.2	Routing algorithms, Shortest path Routing	1
3.2.1	flooding & Distance vector routing	1
3.2.2	link state routing & hierarchical routing	1
3.3	congestion control principle	1
3.4	connectionless internetworking	1
3.5	IP protocol and address	1
3.6	Internet Control Protocols,IPV6	1
3.7	Header Extension headers & limitations – **Tutorial**	1
4	**Transport Layer and DNS**	**(9)**

4.1	Transport Service and its primitives	1
4.2	User Datagram Packets	1
4.3	Remote Procedure Call	1
4.4	TCP Service Model	1
4.5	TCP Protocol and Segment Header	1
4.6	TCP connection establishment and release	1
4.7	modelling TCP Connection Management	1
4.8	TCP congestion control and Timer Management	1
4.9	Domain Name Services & World Wide Web – **Tutorial**	1
5	**Quality of Services**	**(7)**
5.1	Quality of Service in IP Networks,	1
5.2	Integrated and Differentiated services,	1
5.3	Queuing Discipline and Random Early detection	1
5.4	differentiated services	1
5.5	Protocols for QOS support	1
5.6	resource reservation protocol	1
5.7	Introduction to Storage Networks – **Tutorial**	1
	Total	**40**

Course Designers:

1. K. Narasimha Mallikarjunan arjunkambaraj@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT15	3	1	-	4

CT15 Compiler Design 3:1

Preamble: This course gives an overview of the various phases of compiler and explains how a high level program is translated 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 generation of semantic rules during syntax directed translation phase , intermediate code generation and code optimization.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	10	10	10
2	Understand	30	20	20
3	Apply	60	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 analyser?
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.

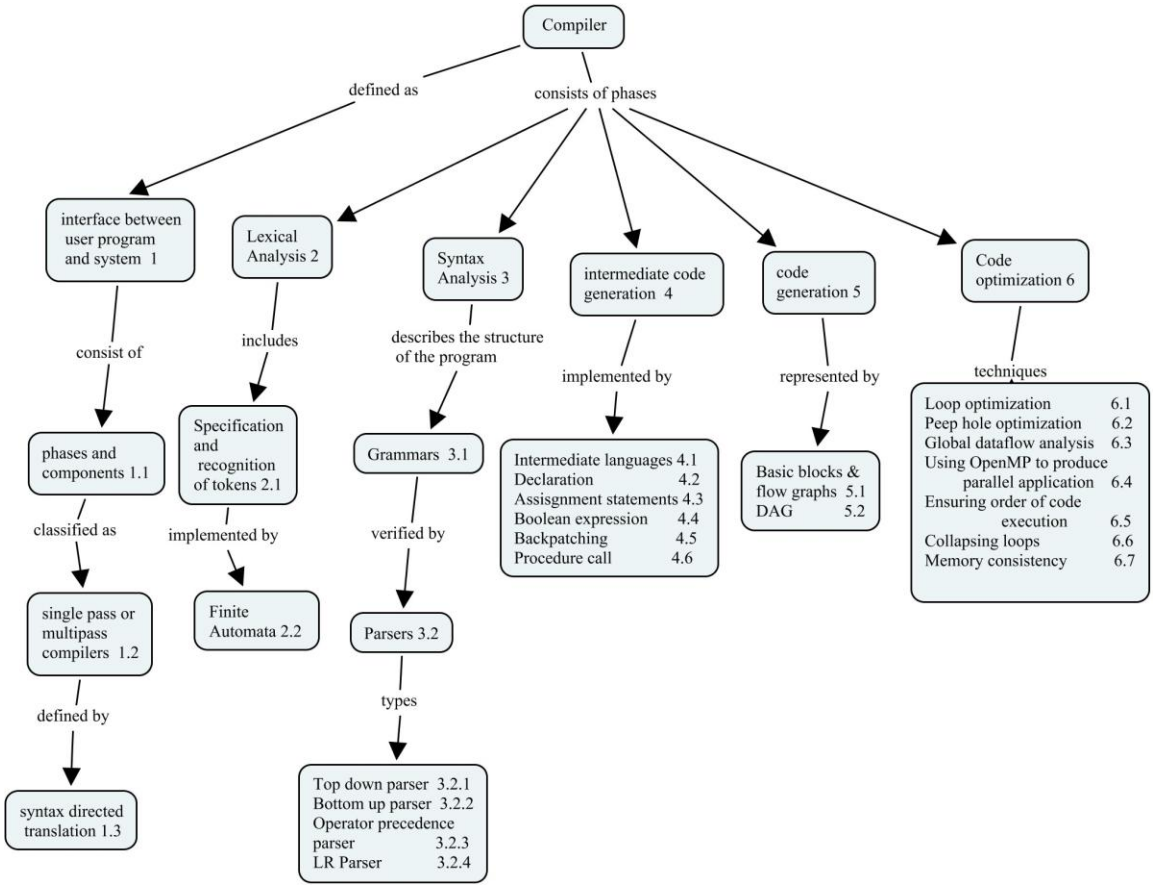
   E→E+T | T

   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 (5)**	
1.1	Phases and components of a compiler	1
1.2	One pass compiler	2
1.3	Syntax-directed translation	2
2.	**Lexical Analysis (8)**	
2.1	Specification and recognition of tokens	3
2.1.1	Design of Lex and Yacc – Tutorial	1
2.2	Finite Automata	3
2.2.1	Tutorial	1
3.	**Syntax Analysis (9)**	

3.1	Grammar	1
3.2	Parsers	1
3.2.1	Top-down Parsing – Tutorial	2
3.2.2	Bottom-up parsing	1
3.2.3	Operator-precedence parsing	2
3.2.4	LR – SLR parser,CLR Parser, LALR parser – Tutorial	2
4.	**Intermediate Code Generation(6)**	
4.1	Intermediate Language	1
4.2	Declaration	1
4.3	Assignment statement	1
4.4	Boolean Expression	1
4.5	Back patching – Tutorial	1
4.6	Procedure Call	1
5.	**Code Generation (3)**	
5.1	Basic blocks and flow graphs – Tutorial	1
5.2	The DAG representation	2
6	**Introduction to Code Optimisation (9)**	
6.1	Loop optimization – Tutorial	1
6.2	Peephole Optimization	1
6.3	Global data flow analysis	1
6.4	Using OpenMP to produce a parallel application	1
6.5	Ensuring that code in a parallel region is executed in order	1
6.6	Collapsing loops to improve workload balance	1
6.7	Enforcing memory consistency	1
6.8	An example of parallelization	2
	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, CLR parser, LALR parser. **Intermediate Code Generation:** intermediate languages, declarations, assignment statements, Boolean expression, procedure call, Back patching **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 using OpenMP to produce parallel application , ensuring code in parallel environment , collapsing loops to improve workload balance, enforcing memory consistency, an example of parallelization

References:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D Ullman, Lam – Compiler Principles, Techniques and Tools, Pearson Education, 2007.
2. Darryl Gove - Multicore Application programming: For Windows,Linux and oracle solaris, Addison Wesley Publisher , 2007
3. Steven S. Muchnick: Advanced Compiler Design & Implementation – Harcourt Asia, Morgan Kaufmann,2001.
4. J. P. Bennet: Introduction to Compiling Techniques, Tata McGraw-Hill Publishing-2002.

Course Designers

1. **B. Thanga Parvathi btpcse@tce.edu**
2. **R.Chellamani rcmcse@tce.edu**

Sub Code	Lectures	Tutorial	Practical	Credit
CT 16	3	1	-	4

CT16 Graphs and Combinatorial Algorithms 3:1

Preamble: Combinatorial reasoning underlies all analysis of computer systems. Two of the most basic mathematical aspects of computer science concern the speed and logical structure. Speed involves enumeration of the number of times each step in a program can be performed. Logical structure involves flow charts, a form of graphs. In graph theory, combinatorial arguments are made a little easier by the use of pictures of the graphs. Natural form of graphs is sets with logical or hierarchical sequencing, such as computer flow charts. In case of network flows, we shall see that the flow optimization algorithm can also be used to prove several well-known combinatorial theorems. The general counting methods involve permutations and combinations. These methods are very useful in constructing computer programs and in mastering many theoretical topics of computer science. Recurrence relations are one of the simplest ways to solve counting problems. The methods for solving recurrence relations appeared originally in the development of the theory of difference equations, cousins of differential equations.

Competencies:

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

1. Verify the pairs of graphs which are isomorphic.
2. Calculate the regions of the given plane graph.
3. Examine the graph whose adjacency matrix is given below to see if is connected.
4. Apply the Augmenting flow Algorithm and calculate the maximal flow for the given graph
5. Determine the number of arrangements and selections using the basic counting principles.
6. Determine the recurrence relation.
7. Determine the solutions of the recurrence relation.

Assessment Pattern

Sl.No.	Bloom's	Test I	Test II	Test III / End
1.	**Remember**	30	30	10
2.	**Understand**	0	0	20
3.	**Apply**	70	70	70

4.	**Analyze**	0	0	0
5.	**Evaluate**	0	0	0
6.	**Create**	0	0	0

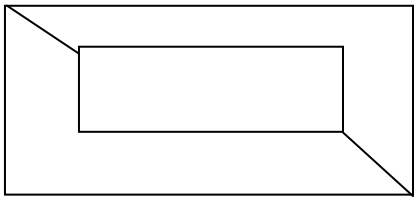
Course Level Learning Objectives

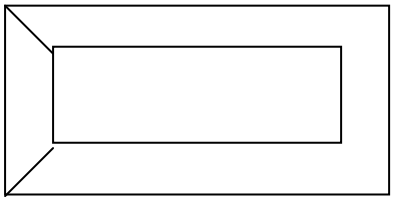
Remember

1. Define complete bipartite graph.
2. State Kruskal's algorithm.
3. Describe Edge- disjoint paths in a graph
4. State two basic counting principles.
5. Define recurrence relation.

Understand

1. State and prove Euler's formula.
2. Write the two algorithms for building a minimal spanning tree.
3. Explain Dijkstra's algorithms with an example
4. Write the Augmenting flow Algorithm and explain it.
5. Are the two graphs in the following figure isomorphic? Justify.





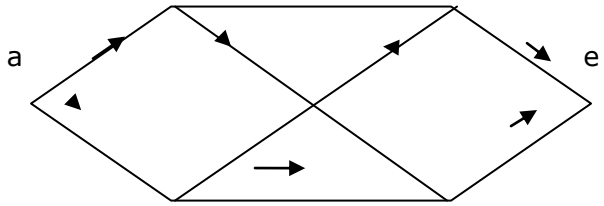
6. Explain Divide and Conquer relation in the analysis of recursive algorithms.

Apply

1. Show that an undirected graph has an Euler cycle if and only if it is connected and has all the vertices of even degree.
2. Examine the graph whose adjacency matrix is given below to see if is connected.

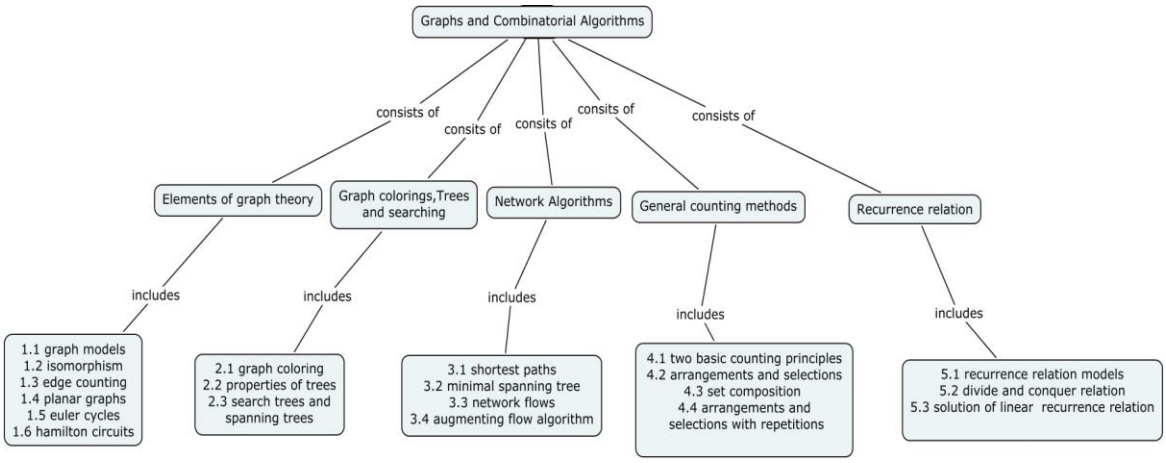
	X1	X2	X3	X4	X5	X6	X7	X8
X1	0	1	1	1	0	1	0	0
X2	1	0	1	0	1	0	1	0
X3	1	1	0	0	0	0	0	0
X4	1	0	0	0	0	0	1	0
X5	0	1	0	0	0	0	1	0
X6	1	0	0	0	0	0	0	0
X7	0	1	0	1	1	0	0	1
X8	0	0	0	0	0	0	1	0

3. Apply the Augmenting flow Algorithm and calculate the maximal a - e flow for the graph given below.



4. What is the probability that a 4- digit campus telephone number has one or more repeated digits? And determine the probability that,
   i. (i). All 4 digits are the same.
   ii. (ii). 3 digits are the same, the other is different.
   iii. (ii). 2 digits are the same, the other 2 digits are each different.
5. Solve the recurrence relation $a_n = 2a_{n-1} + 3a_{n-2}$ with $a_0 = a_1 = 1$.
6. Determine how many ways are there to arrange the seven letter in the word SYSTEMS? In how many of these arrangements do the 3 S's appear consecutively?
7. Enumerate all arrangements of a, c, e, h in lexicographic order.

Concept map



Syllabus : Module-I: Elements of graph theory Graph models, isomorphism, Edge counting, Planar graphs, Euler cycles, Hamilton circuits. **Module-II: Coloring, trees and searching** Graph coloring, Chromatic polynomial, Properties of Trees, Cayley's theorem, Depth first and breadth first search, Spanning Trees**. Module-III: Network Algorithms** Shortest paths Algorithms , Minimal spanning trees, Network flows , Augmenting flow Algorithm, Max flow- Min cut theorem, Undirected networks-flow networks with supplies and demands . **Module-IV: Counting principles** Two basic counting principles , Simple Arrangements and selections, Set composition principle, Arrangements and selections with Repetitions , Generating permutation and

combinations. **Module-V: Recurrence Relations** Recurrence Relations Models, Fibonacci number, Divide and conquer Relations, Solution of linear Recurrence Relations, Solution of in Homogeneous Recurrence Relations.

References:

1. Alan Thucker, "Applied Combinatories " , John Wiley and Sons, Inc, New York, Third Edition,1995.

2. CL Lin, " Introduction to Combinatorial Mathematics", Mcgraw Hill Book Company, New York, 1968.

3. Richard A.Brualdi, "Introductory Combinatories", North Holland, New York

Course Contents and Lecture schedule

S.No	Topics	No. of Lectures
	Module I Elements of graph theory (9)	
1	Review of the concepts of Set theory, relations & functions	1
2	Graph models, Isomorphism	2
3	Edge counting	1
4	Planar graphs	2
5	Euler cycles	1
6	Hamilton circuits	1
7	Tutorial	1
	Module-II Coloring , trees and searching(10)	
8	Graph coloring	2
9	Chromatic polynomial	1
10	Properties of Trees	2
11	Cayley's theorem	1
12	Depth first and breadth first search	2
13	Spanning Trees	1
14	Tutorial	1
	Module-III Network Algorithms(7)	
15	Shortest paths Algorithms	1
16	Minimal spanning trees	1
17	Network flows	1
18	Augmenting flow Algorithms	1
19	Max flow- Min cut theorem	1
20	Undirected networks-flow networks with supplies and demands	1
21	Tutorial	1

	Module-IV Counting principles(6)	
22	Two basic counting principles	1
23	Simple Arrangements and selections	1
24	Set composition principle	1
25	Arrangements and selections with Repetitions	1
26	Generating permutation and combinations	1
27	Tutorial	1
	Module-V Recurrence Relations(8)	
28	Recurrence Relations Models	1
29	Fibonacci number	1
30	Divide and conquer Relations	1
31	Solution of linear Recurrence Relations	2
32	Solution of in Homogeneous Recurrence Relations	2
33	Tutorial	1
	Total	**40**

Course Designers

1. V.Mohan vmohan@tce.edu
2. S.Jeya Bharathi sjbmat@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT 17	-	-	3	1

CT17 Analysis and Design Laboratory 0:1

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

List of Exercises:

1. Implementation of hash function
2. Implementation of Linked list, doubly linked list and Circular linked list.
3. Implementation of sorting and searching algorithms.
4. Implementation of Binary search tree, Red Black tree.
5. Implementation of graph traversal techniques.
6. Implementation of Use Case Diagram
7. Implementation of Class Diagram
8. Implementation of State Diagram
9. Implementation of Activity Diagram
10. Implementation of Sequence Diagram
11. Implementation of Collaboration Diagram
12. Implementation of Deployment Diagram

Course Designer

1. J. Jane Rubel Angelina janerubel@tce.edu
2. S. Padmavathi spmcse@tce.edu

Annexure – EE

CURRICULUM AND DETAILED SYLLABI

FOR

M.E DEGREE (Computer Science and Engineering) PROGRAMME

SECOND SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2011-2012 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

Department of Computer Science and Engineering

Graduating Students of ME 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 the M.E.CSE (for those who are admitted from the year 2011-2012)

4th (12)							Project Phase II 0:12
3rd (16)	CT31 Data Warehousing and Mining Techniques 4:0	CTX Elective 5 4:0	CTX Elective 6 4:0				Project Phase I 0:4
2nd (24)	CT21 Modeling and Simulation 3:1	CT22 Software Engineering Theory and Practice 3:0	CTX Elective 1 4:0	CTX Elective 2 4:0	CTX Elective 3 4:0	CTX Elective 4 4:0	Seminar 0:1
1st (24)	CT11 Computer Architecture 4:0	CT12 Data Structures and Algorithms 3:1	CT13 Object Oriented Analysis and Design 3: 0	CT14 Network Technology 3:1	CT15 Compiler Design 3:1	CT16 Graphs and Combinatorial Algorithms 3:1	CT17 Analysis and Design Lab 0:1

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

(An Autonomous Institution affiliated to Anna University)

CURRICULAM

(For the Students admitted from the academic year 2011-2012)

Name of the Degree: M.E (Computer Science and Engineering) Programme

SUBJECTS OF STUDY

I SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT11	Computer Architecture	4	0	0	4
CT12	Data Structures and Algorithms	3	1	0	4
CT13	Object Oriented Analysis and Design	3	0	0	3
CT14	Network Technology	3	1	0	4
CT15	Compiler Design	3	1	0	4
CT16	Graphs and Combinatorial Algorithms	3	1	0	4
Practical:					
CT17	Analysis and Design Laboratory	0	0	3	1
		Total Credits			**24**

II SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT21	Modeling and Simulation	3	1	0	4
CT22	Software Engineering Theory and Practice	3	0	0	3
CTX	Elective 1	4	0	0	4
CTX	Elective 2	4	0	0	4
CTX	Elective 3	4	0	0	4
CTX	Elective 4	4	0	0	4
Practical:					
CT27	Seminar	0	0	3	1
		Total Credits			**24**

III SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT31	Data Warehousing and Mining Techniques	4	0	0	4
CTX	Elective 5	4	0	0	4
CTX	Elective 6	4	0	0	4
Practical:					
CT34	Project	0	0	12	4
		Total Credits			**16**

IV SEMESTER

Practical:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT41	Project	0	0	36	12
		Total Credits			**12**

Total No. of credits to be earned for the award of degree: 76

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E Degree (Computer Science and Engineering) Programme
SCHEME OF EXAMINATIONS
(For the candidates admitted from 2011-2012 onwards)

I 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	CT11	Computer Architecture	3	50	50	100	25	50
2	CT12	Data Structures and Algorithms	3	50	50	100	25	50
3	CT13	Object Oriented Analysis and Design	3	50	50	100	25	50
4	CT14	Network Technology	3	50	50	100	25	50
5	CT15	Compiler Design	3	50	50	100	25	50
6	CT16	Graphs and Combinatorial Algorithms						
PRACTICAL								
7	CT17	Analysis and Design Laboratory	3	50	50	100	25	50

II 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	CT21	Modeling and Simulation	3	50	50	100	25	50
2	CT22	Software Engineering Theory and Practice	3	50	50	100	25	50

3	CTX	Elective 1	3	50	50	100	25	50
4	CTX	Elective 2	3	50	50	100	25	50
5	CTX	Elective 3	3	50	50	100	25	50
6	CTX	Elective 4	3	50	50	100	25	50
PRACTICAL								
7	CT27	Seminar	3	50	50	100	25	50

III 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	CT31	Data Warehousing and Mining Techniques	3	50	50	100	25	50
2	CTX	Elective 5	3	50	50	100	25	50
3	CTX	Elective 6	3	50	50	100	25	50
PRACTICAL								
4	CT34	Project	-	150	150	300	75	150

IV 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	CT41	Project	-	150	150	300	75	150

Sub Code	Lectures	Tutorial	Practical	Credit
CT21	3	1	-	4

CT21 Modeling and Simulation 3:1

Preamble Modeling and Simulation refers analysis tool and design tool for various systems. 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 addition to this, random variate for various distributions is also generated. In analysis of simulation, input values are generated and used for experiment. It includes the verification and validation of simulation model. Finally the modeling and simulation concept is applied in CPU simulation, memory simulation

Prerequisite Probability and Statistics, Queuing theory

Program Outcomes Addressed

a. Graduate will demonstrate the knowledge of Mathematics, science and Engineering
b. Graduate will demonstrate an ability to identify , formula and solve the problems
c. 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. 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. What is simulation tool?
5. What is goodness of fit?
6. What is validation

Understand

1. Write the application of simulation
2. What is CPU simulation?
3. Write the inverse transform technique
4. What is objective of frequency test
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. Test for whether the 3rd, 8th, 13th and so on numbers given below are auto correlated using $\alpha = 0.05$

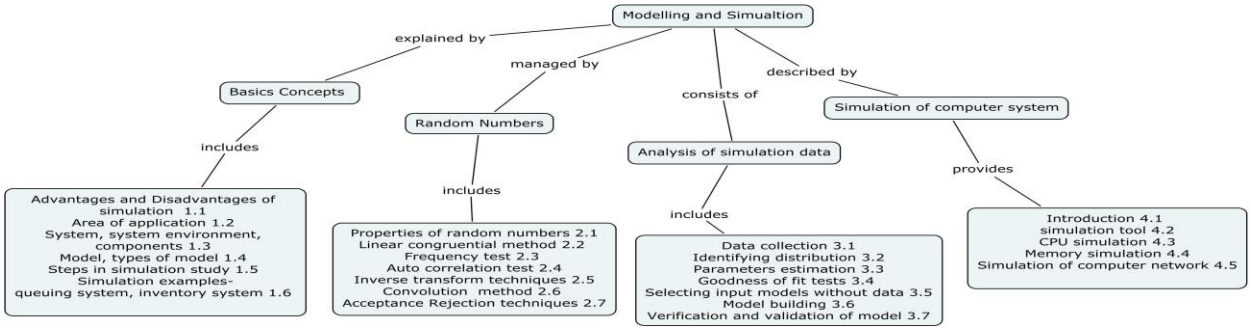
0.41 0.68 0.89 0.94 0.74 0.91 0.55 0.62 0.36 0.27 0.19 0.72 0.75 0.08 0.54 0.02 0.01 0.36 0.16 0.28 0.18 0.01 0.95 0.69 0.18 0.47 0.23 0.32 0.82 0.53 0.31 0.42 0.73 0.04 0.83 0.45 0.13 0.57 0.63 0.29

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: Simulation is the appropriate tool, not appropriate tool, advantages and disadvantages of simulation, Area of application, System, system environment, components, Model, types of model, Steps in simulation study, Simulation examples- queuing system, inventory system. **Random Numbers:** Properties of random numbers, Pseudo random numbers, linear congruential method, Frequency test, Auto correlation test, random variate generation-Inverse transform technique, convolution method, Acceptance Rejection technique, **Analysis of simulation data**: input modeling- Data collection, identifying the distribution with data, Parameters estimation, Goodness of fit tests. Selecting input models without data, Model building, Verification and validation of simulation models **Simulation of computer system**: Introduction, simulation tool, CPU simulation, memory simulation, simulation of computer network

References

1. Jerry Bank, John S. Carson II, Barry L. Nelson, David M. Nicol, P. Shahabudeen , " Discrete Event System Simulation ", Pearson, 2007

2. Narasingh Deo, "System Simulation with Digital Computer", PHI, New Delhi, 1997
3. 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	Simulation is the appropriate tool, not appropriate tool, advantages and disadvantages of simulation	**2**
1.2	Area of application	**1**
1.3	System, system environment, components	**1**
1.4	Model, types of model	**1**
1.5	Steps in simulation study	**1**
1.6.	Simulation examples-queuing system, inventory system	**4**
1.7	Tutorial	**1**
2	**Random Numbers**	
2.1	Properties of random numbers, Pseudo random numbers	**1**
2.2	Linear congruential method	**1**
2.3	Frequency test	**3**
2.4	Tutorial	**1**
2.5	Auto correlation test	**2**
2.6	Inverse transform techniques	**4**
2.7	Convolution method	**1**
2.8	Acceptance Rejection techniques	**3**
2.9	Tutorial	**1**
3	**Analysis of simulation data**	
3.1	Introduction, Data collection	**1**
3.2	Identifying distribution	**1**
3.3	Parameters estimation	**3**
3.4	Goodness of fit tests	**4**
3.5	Tutorial	**1**
3.6	Selecting input models without data	**1**
3.7	Model building	**1**

3.8	Verification and validation of simulation models	**3**
3.9	Tutorial	**1**
4	**Simulation of computer system**	
4.1	Introduction	**1**
4.2	simulation tool	**1**
4.3	CPU simulation	**2**
4.4	Tutorial	**1**
4.5	Memory simulation	**2**
4.6	Simulation of computer network	**5**
4.7	Tutorial	**1**
	Total number of hours	**50**

Course Designers

1. Dr.V.Mohan(vmohan@tce.edu)

2. Mr.M.Mutharasan(mmmat@tce.edu)

Sub Code	Lectures	Tutorial	Practical	Credit
CT22	3	-	-	3

CT22 Software Engineering Theory and Practice 3:0

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 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 be able to work on multidisciplinary teams.

Competencies

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

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

Course Level 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.

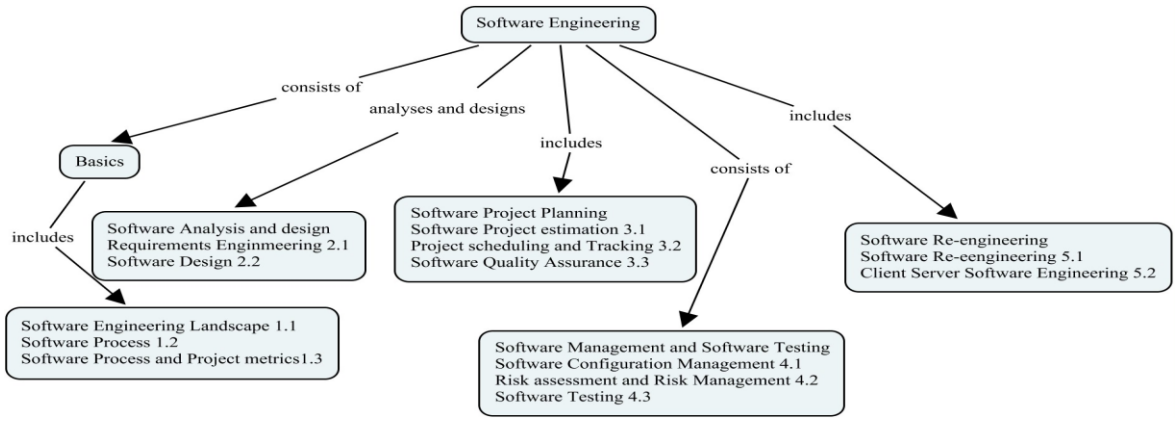
```
/*  SelectionSorter.java */

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



Syllabus

Software Engineering Landscape – Software engineering components – Software development problems – CMM – Software standards. **Software Process** – ETVXM Architecture – Profiles – Development Process – Lifecycle model – Clean room process model – Universal software process model. **Software Process and Project Metrics** –

Measures, Metrics and Indicators – Software measurement – Metrics for Software Quality.**Software Analysis: Requirements Engineering** – Problem Analysis – Software Requirements Specification – Object Oriented Requirement Analysis – Function Oriented Analysis - Development Methods. **Software Design: Architecture** – Software Architectures – Architectural Styles – Families of Software Architecture. **Software Project Planning -** Software Project Estimation – Decomposition Techniques- Empirical Estimation Models- Make/Buy Decision – Automated Estimation Tools. **Project Scheduling and Tracking –** Defining a task network – Scheduling – Earned Value Analysis- Error Tracking. **Software Quality Assurance** – Quality Concepts – Quality Movement – Software Quality Assurance – Software Reviews – Formal Technical Reviews – Statistical Software Quality Assurance.**Software Configuration Management** – Identification – Version Control – Change Control – Configuration Audit – Status Reporting – SCM Standards. **Risk Analysis and Risk Management** – Risk Strategies – Software Risks- Risk Identification – Risk Projection – Risk Refinement – RMMM. **Software Testing** – Software Testing Fundamentals – Test Case Design – White Box Testing – Basis Path Testing – Control Structure testing - Black Box 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

References

1. James F. Peters , "Software Engineering – An Engineering Approach", John Wiley & Sons. 2004.

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

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

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

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

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

Course Content and Lecture Schedule

No	Topic	No of Lectures
1.	**Software Engineering**	

1.1	**Software Engineering Landscape**	
1.1.1	Software engineering components , Software development problems	1
1.1.2	CMM, Software standards	1
1.2	**Software Process**	
1.2.1	ETVXM Architecture	2
1.2.2	Profiles, Development Process	1
1.2.3	Lifecycle model	1
1.2.4	Clean room process model	1
1.2.5	Universal software process model.	1
1.3	**Software Process and Project Metrics**	
1.3.1	Measures, Metrics and Indicators,	1
1.3.2	Metrics for Software Quality.	1
2.	**Software Analysis and Design**	
2.1	**Requirements Engineering**	
2.1.1	Software Requirements Specification	1
2.1.2	Object Oriented Requirement Analysis	1
2.1.3	Function Oriented Analysis	1
2.1.4	Development Methods	1
2.2	**Software Design**	
2.2.1	Software Architectures	1
2.2.2	Architectural Styles	1
2.2.3	Families of Software Architecture	1
3.	**Software Project Planning**	
3.1	**Software Project Estimation**	
3.1.1	Decomposition Techniques	1
3.1.2	Empirical Estimation Models	1
3.1.3	Make/Buy Decision, Automated Estimation Tools	1

3.2	**Project Scheduling and Tracking**	
3.2.1	Defining a task network	1
3.2.3	Scheduling, Earned value analysis, Error Tracking	1
3.3	**Software Quality Assurance**	
3.3.1	Quality Concepts, Quality Movement	1
3.3.2	Software Quality Assurance, Software Reviews	1
3.3.2	Formal Technical Reviews, Statistical Software Quality Assurance	1
4	**Software Management and Testing**	
4.1	**Software Configuration Management**	
4.1.1	Identification, Version Control	1
4.1.2	Change Control	1
4.1.3	Configuration Audit, Status Reporting, SCM Standards	1
4.2	**Risk Analysis and Risk Management**	
4.2.1	Risk Strategies, Software Risks	1
4.2.2	Risk Identification, Risk Projection	1
4.2.3	Risk Refinement, RMMM	1
4.3	**Software Testing**	
4.3.1	Software Testing Fundamentals, Test Case Design	1
4.3.2	White Box Testing, Basis Path Testing	1
4.3.3	Control Structure testing	1
4.3.4	Black Box Testing	1
5	**Reengineering**	
5.1	**Software Reengineering**	
5.1.1	Business process reengineering	1
5.1.2	Software Reengineering, Reverse Engineering	1
5.1.3	Restructuring, Forward Engineering	1

5.1.4	Reengineering Economics	1
5.2	**Client / Server Software Engineering**	1
	Total number of hours	**40**

Course Designer

1. Dr. A.Askarunisa aacse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTA	3	1	-	4

CTA Database Systems 3:1

Preamble This course aims at facilitating the student to understand the concepts of database systems, design 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 identify, formulate and solve engineering problems.

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

d. 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 the database with constraints and retrieve data using queries.
3. Design a database using normalization.
4. Retrieve the data from database by applying various access methods.
5. Manage the database using transaction and concurrency control mechanisms.
6. Understand the concepts of distributed database systems.
7. Understand the concepts of different applications of databases including spatial, temporal and image

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	20	20	30
3	Apply	50	40	40
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	10	20	20

Course Level Learning Objectives

Remember

1. What is the purpose of data dictionary?

2. Define ACID properties.
3. List the different states of a transaction?
4. What is dense and sparse index?
5. What are the advantages of distributed database?
6. What are the new data types used in spatial application?
7. What is the need for normalization?

Understand

1. Differentiate static and dynamic hashing.
2. Why BCNF is stronger than 3NF? Justify.
3. How do you implement atomicity in transactions?
4. Compare and contrast object oriented and object relational data models.
5. How might a distributed database designed for local area network differ from one designed for wide area network?
6. Explain different spatial operations used in spatial queries?
7. Explain the distinction between serial schedule and serializable schedule.

Apply

1. Consider the following schema and write the following queries in relational algebra.

   Suppliers (sid, sname, address)
   Parts (pid, pname,color)
   Catelog (sid,pid,cost)

   a. Find the names of suppliers who supply some red part.
   b. Find the sids of suppliers who supply some red and green part.
   c. Find the sids of suppliers who supply every part.
   d. Find the pids of the most expensive parts supplied by supplier named Sham

2. Consider the following schema and write the SQL for the following:

   Country(name,cont,pop,gdp,life-exp)
   River(name,origin,lenth)
   City(name, country, pop)

   a. Find all countries 8whose GDP is greater than $700 billion but less than $3 trillion.
   b. List the life expentency in countries that have river originating in them
   c. Find all cities that are either in south america or whose population is less than 4 million.

d. List all cities which are not in south America

3. Consider the following relation:

X	**Y**	**Z**
x1	y1	z1
x1	y1	z2
x2	y1	z1
x2	y1	z3

a. List all the functional dependencies that this relation instance satisfies.
b. Assume that the value of attribute Z of the last record in the relation is changed from z3 to z2. Now list all the functional dependencies that this relation instance satisfies.

4. Consider a relation R with five attributes ABCDE. You are given the following dependencies: A→B, BC→E, and ED→A.
   a. List all keys for R.
   b. Is R in 3NF?
   c. Is R in BCNF?

5. Consider the database that includes the following relations:

   Salaried_worker(name, office, phone, salary)
   Hourly_worker (name, hourly_wage)
   Address(name, street, city)

   Suppose that we want to require that every name that appears in address appears in either salaried_worker or hourly_worker, but not necessarily in both.
   a. Propose a syntax for expressing such constraitns.
   b. Discuss the actions that the system must take to enforce a constraitn for this form.

6. Consider the following two 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);
```

Let the consistency requirement be A = 0 V B = 0, with A = B = 0 the initial values.

a. Show that every serial execution involving these two transactions preserves the consistency of the database.
b. Show a concurrent execution of T1 and T2 that produces a non serializable schedule.

Create

1. Design a generalization-speciation hierarchy for a motor vehicle sales company. The company sells motor cycles, passenger cars, vans, and buses. Identify the attributes at each level of the hierarchy.

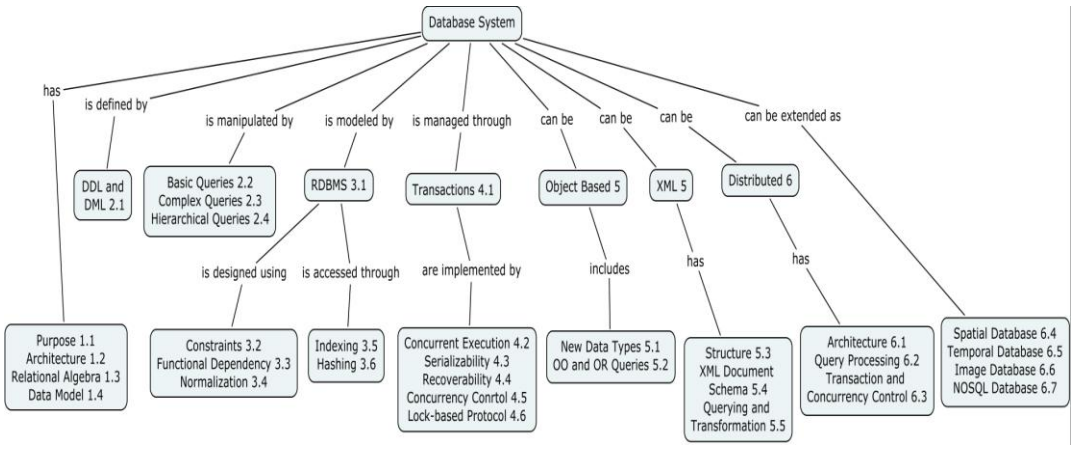
2. For the following employee database

   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.
3. Construct a B+ tree with order 3 to insert the following key elements:

   55, 32, 12, 5, 7, 80, 76, 33, 61, 90, 2
4. Consider a database used to record the marks that students get in different exams of different course offerings. Construct an E-R diagram that models exams as entities, and uses a ternary relationship for the database.
5. Suppose that we are using extendable hashing on a file that contains records with the following search-key values: 2, 3, 5, 7, 11, 17, 19, 23, 29, and 31. Show the extendable hash structure for this file if the hash function is h(x) = x mod 8 and buckets can hold three records.
6. Design the database for banking environment by following the design phases including normalization.

Concept Map



Syllabus

Introduction: Purpose, Architecture, Relational Algebra, Data Model. **Structured Query Language**: DDL and DML, Basic Queries, Complex Queries, Hierarchical Queries **Relational Database Design:** RDBMS Concepts, Constraints, Functional Dependency, Normalization, Indexing, Hashing. **Transaction Management:** Transaction – Concepts, Concurrent Execution, Serializability, Recoverability, Concurrency Control, and Lock based Protocol. **Object Based Databases and XML:** New Data Types in SQL, Object Oriented and Object Relational Queries, Structure of XML Data, XML Document Schema, Querying and Transformation. **Distributed Database and New Applications:** Distributed Database – Architecture, Query Processing, Transaction and Concurrency Control, Spatial Database, Temporal Database, Image Database and NOSQL database.

References

1. Silberschatz, Korth, Sudarshan, "Database System Concepts", 5th Edition, Tata McGrawHill, 2006.
2. Thomas Connolly, Carolyn Begg, "Database Systems – A practical Approach to Design, Implementation and Management", 3rd Edition, Pearson Eduction, 2003.
3. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database System", 4th edition, Pearson Education, 2004.
4. Raghu Ramakrishnan and Johannes Gehrke, "Database Management System", 3rd edition, TMH, 2003.
5. Shashi Shekhar and Sanjay Chawla, "Spatial Database – A Tour", Pearson Education, 2003.

Course Contents and lecture schedule

No.	Topic	No. of Lectures
1	**Introduction**	
1.1	Purpose	1
1.2	Architecture	1
1.3	Relational Algebra	2
1.4	Data Model	2
2.	**Structured Query Language**	
2.1	DDL and DML	1
2.2	Basic Queries	1

No.	Topic	No. of Lectures
2.3	Complex Queries	2
2.4	Hierarchical Queries	1
3	**Relational Database Design**	
3.1	RDBMS - Concepts	1
3.2	Constraints	1
3.3	Functional Dependency	1
3.4	Normalization	2
3.5	Indexing	2
3.6	Hashing	2
4	**Transaction Management**	
4.1	Transactions – Concepts	1
4.2	Concurrent Execution	1
4.3	Serializability	2
4.4	Recoverability	1
4.5	Concurrency Control	1
4.6	Lock-based protocol	1
5	**Object Based Databases and XML**	
5.1	New Data Types in SQL	1
5.2	Object Oriented and Object Relational Queries	1
5.3	Structure of XML Data	1
5.4	XML Document Schema	1
5.5	Querying and Transformation	1
6	**Distributed Database and New Applications**	
6.1	Distributed Database - Architecture	1
6.2	Query Processing	1
6.3	Transaction and Concurrency Control	1
6.4	Spatial Database	2

No.	Topic	No. of Lectures
6.5	Temporal Database	1
6.6	Image Database	1
6.7	NOSQL Database	1
	Total	**40**

COURSE DESIGNERS:

1. Mr. M. Sivakumar mskcse@tce.edu
2. Ms. B. Subbulakshmi bscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTB	4	0	-	4

CTB Information Storage and Networking 4:0

Preamble The course on Information Storage and Networking aims to emphasize the need for information storage, provide an in depth coverage of various technologies used in the networked storage.

Program Outcomes addressed

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

b. 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 protocols used in the information storage systems
2. Evaluate the existing information storage systems.
3. Create storage system requirement specification document based on interaction with storage user.
4. Investigate the Failure analysis and use of Backup technologies for Recovery.
5. Design and manage the storage infrastructure effectively.

Assessment Pattern

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

Course Level Learning Objectives

Remember

1. What are the different 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. What are the characteristics of Platter?
6. What is TAG RAM?
7. What are the challenges of NAS?

8. Explain DAS performance considerations and challenges.
9. Define: Strip and Stripe.
10. Define Single-mode and Multi-mode fiber.
11. What are the benefits of NAS?

Understand

1. Describe how you can control Application access, User access and Host access.
2. Explain how CAS stores and retrieves data objects
3. Write down the benefits of CAS.
4. Describe the disk drive components and its operation in detail
5. Describe the evolution of date storage systems.
6. Explain the structure of Cache
7. Describe the functionalities of FC Switch and Hub with necessary diagrams.
8. Write down the benefits and forms of Virtualization
9. Describe the disk drive components and its operation in detail
10. Compare and contrast NFS and CIFS.
11. Differentiate between Internal and External DAS management
12. Compare and contrast between Integrated and Gateway NAS
13. Explain the IP SAN protocol
14. Compare the different storage media with their merits and demerits
15. Explain how remote replication technology is helpful in disaster recovery.
16. Distinguish Strip from Stripe.
17. Compare Single-mode and Multi-mode fiber.

Apply

1. An application specifies a requirement of 200GB to host a database and other files. It also specifies that the storage environment should support 5000 IOPS during its peak processing cycle. The disks available for configuration provide 66GB of usable capacity and the manufacturer specifies that they can support a maximum of 140 IOPS. The application is response time sensitive and the disk utilization beyond 60% will not meet the response time requirements of the application. Compute the minimum number of disks that should be configured to meet the requirements of the application.
2. Consider a disk I/O system in which an I/O request arrives at the rate of 80 IOPS. The disk service time is 6ms.
    a. Compute the following
        i. Utilization of I/O controller
        ii. Total response time
        iii. Average queue size
        iv. Total time spent by a request in a queue
    b. Compute the preceding parameter if the service time is halved.

3. A 10k RPM drive is rated to perform 130 IOPS and a 15k RPM drive is rated to perform 180 IOPS for an application. The read/write ratio is 3:1. Compute the RAID-adjusted IOPS for the 10k and 15k drives for RAID 1, RAID 5 and RAID 6.
4. An application has 1000 heavy users at a peak of 2 IOPS each and 2000 typical users at a peak of 1 IOPS each, with a read/write ratio of 2:1. It is estimated that the application also experiences an overhead of 20% for other workloads. Calculate the IOPS requirement for RAID 1, RAID 3, RAID 5 and RAID 6.
5. A UNIX host has a path to a storage device that shows as c0 t1 d3. Draw a diagram to show the path.
6. Seventeen switches with 16 ports each are connected in a mesh topology. Determine the ports available for host and storage connectivity for a high-availability solution.
7. SAN is configured for a backup to disk environment and the storage configuration has additional capacity available. Determine whether a NAS gateway can be configured to utilize this SAN.
8. 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. Determine a NAS solution.
9. The IP bandwidth provided for FCIP connectivity seems to be constrained. Estimate the implications if the SANs that are merged are fairly large with 500 ports on each side and the SANs at both the ends are constantly reconfigured.

Create

1. AirTel Telecom is involved in mobile wireless services across the India and has about 5000 employees worldwide. This company is Chennai based and has 7 regional offices across the country. Although AirTel is doing well financially, they continue to feel competitive pressure. As a result, the company needs to ensure that the IT infrastructure takes advantage of fault tolerant features.

   Current Situation/Issues

   i. The company uses a number of different applications for communication, accounting, and management. All the applications are hosted on individual servers with disks configured as RAID 0.
   ii. All financial activity is managed and tracked by a single accounting application. It is very important for the accounting data to be highly available.
   iii. The application performs around 15% write operations, and the remaining 85 % are reads.
   iv. The accounting data is currently stored on a 5-disk RAID 0 set. Each disk has an advertised formatted capacity of 200 GB, and the total size of their files is 730 GB.

v. The company performs nightly backups and removes old information—so the amount of data is unlikely to change much over the next 6 months. The company is approaching the end of the financial year and the IT budget is depleted. Buying even one new disk drive will not be possible. Design an infrastructure for the company to suit the new requirements. Justify your design based on cost, performance, and availability.

2. 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:

i. Due to the de-centralized backup environment, recoverability of the backup servers is compromised.

ii. Key applications have to be shut down during the backup process.

iii. Too many tapes need to be mounted in order to perform a full recover, in case of a complete failure. The company would like to:

a. Deploy an easy-to-manage backup environment.

b. Reduce the amount of time the email and database applications need to be shutdown.

c. Reduce the number of tapes required to fully recover a server in case of failure.

Create a network based on IP SAN topology.

4. 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.

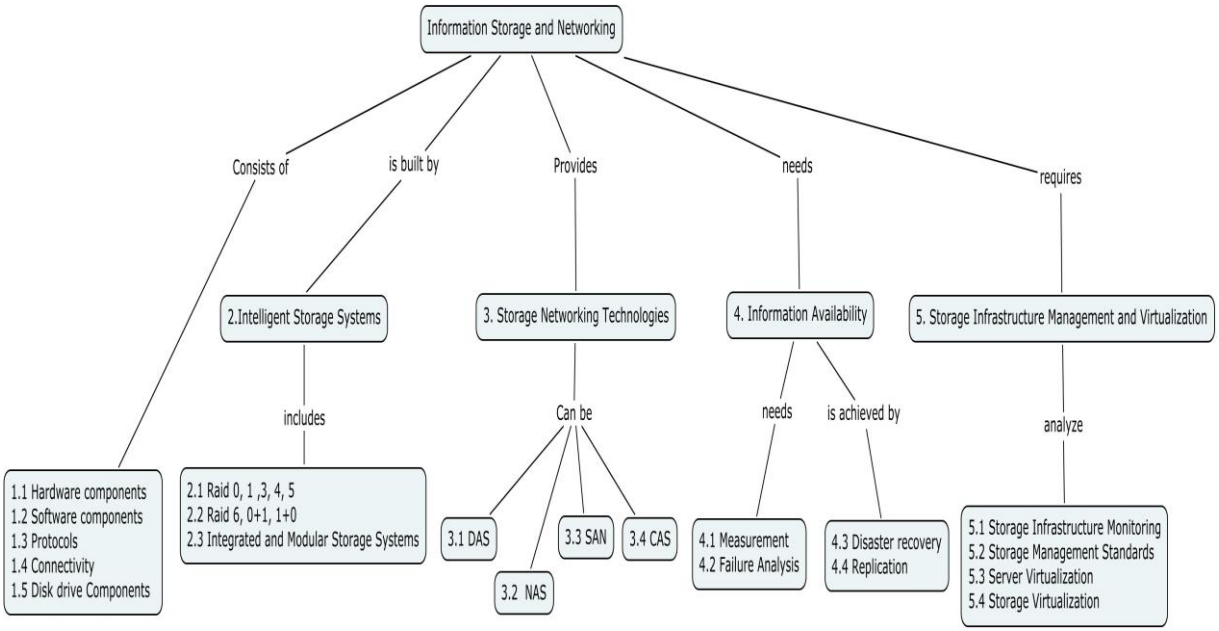
Current Situation/Issue

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.

Develop a better solution based on other remote replication technologies and the zero RPO solution.

6. 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. Create monitoring and management procedures, tools and alerts to ensure accessibility, capacity, performance and security.
7. The Information Department of a departmental store uses tape to archive data. The data once created may be accessed within 30 days and when it crosses that period, the frequency of access is less than 1%. Design a CAS solution.
8. An oracle database uses a block size of 4kb for its I/O operation. The application that uses this database primarily performs a sequential read operation. Design a Cache memory with the following parameters: Cache page size, cache allocation, pre-fetch type and write aside cache.

Concept Map



Syllabus

Introduction - Hardware and software components of the host environment - Key protocols and concepts used by each component - Physical and logical components of connectivity environment - Physical Components of a disk drive - Logical constructs of a physical disk. **Intelligent Storage Systems -** RAID levels: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5 - RAID 6 - RAID 0+1, RAID 1+0 - Integrated and Modular Storage Systems - Intelligent Storage System **Storage Networking Technologies-**Direct Attached Storage :Architecture and components - Network Attached Storage : Architecture and components and connectivity - Storage Area Network :Evolution, Architecture, components and connectivity - Need for long-term archiving solution: CAS :

Object based data storage and Retrieval **Information Availability -**Information Availability and its measurement- causes and consequences of downtime - Failure Analysis: Single points of failure in a storage infrastructure and solutions for its mitigation- Fault Tolerance –Business continuity- Disaster Recovery (DR): Recovery Time Objective (RTO) and Recovery Point Objective (RPO) – Overview of backups : Direct-attached backups- LAN-based backups- LAN-free backups- Server less backups – Archival - Replication - Remote replication: Host-based, Storage Array-based**Storage Infrastructure Management and Virtualization** Storage Infrastructure Monitoring: Parameters, Components - key management activities- storage management standards and initiative - Virtualization technologies: Server Virtualization (LVM –based virtualization, Memory virtualization), Network (VLAN and VSAN), Storage (Disk virtualization, RAID, LUN masking, File and block level virtualization, Virtual Provisioning)

References

1. Information Storage and Management, EMC Corporation, Wiley 2009, ISBN: 04702942134

2. Storage Networks: The Complete Reference - Robert Spalding - Tata McGraw Hill-Osborne- 2003.

3. Building Storage Networks - Marc Farley - Tata McGraw Hill-Osborne- 2001.

4. Disaster Recovery and Business Continuity - Thejendra BS- Shroff Publishers and Distributors-2006.

5. Storage Area Network Fundamentals- Meeta Gupta- Pearson Education Limited, 2002

Course contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Introduction**	
1.1	Hardware components of the host environment	2
1.2	Software components	2
1.3	Key protocols and concepts used by each component	2
1.4	Physical and logical components of Connectivity environment	2
1.5	Physical Components of a disk drive - Logical constructs of a physical disk	2
2	**Intelligent Storage Systems**	
2.1	RAID levels- RAID 0, RAID 1, RAID 3, RAID 4, RAID 5	2
2.2	RAID 0+1, RAID 1+0, RAID 6	2
2.3	Integrated and Modular Storage Systems - Intelligent Storage System	3
3	**Storage Networking Technologies**	
3.1	Direct Attached Storage – Architecture and components	3

No.	Topic	No. of Lectures
3.2	Network Attached Storage – Architecture and components and connectivity	3
3.3	Storage Area Network – Evolution, Architecture, components and connectivity	3
3.4	Need for long-term archiving solutions – CAS : Object based data storage and Retrieval	3
4	**Information Availability**	
4.1	Information Availability and its measurement- causes and consequences of downtime	2
4.2	Failure Analysis: Single points of failure in a storage infrastructure and solutions for its mitigation- Fault Tolerance – Business continuity	3
4.3	Disaster Recovery (DR): Recovery Time Objective (RTO) and Recovery Point Objective (RPO)- Overview of backups : Direct-attached backups- LAN-based backups- LAN-free backups- Server less backups – Archival	3
4.4	Replication - Remote replication: Host-based, Storage Array-based	3
5	**Storage Infrastructure Management and Virtualization**	
5.1	Storage Infrastructure Monitoring: Parameters, Components-key management activities	2
5.2	Storage management standards and initiative	2
5.3	Virtualization technologies: Server Virtualization (LVM –based virtualization, Memory virtualization)	3
5.4	Network (VLAN and VSAN), Storage (Disk virtualization, RAID, LUN masking, File and block level virtualization, Virtual Provisioning)	3
	Total number of hours	50

COURSE DESIGNER

1. Mr. M. P. Ramkumar ramkumar@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTC	4	-	-	4

CTC MACHINE LEARNING **4:0**

Preamble: This course will present the key algorithms and theory that form the core of machine learning. Since machine learning draws on concepts and results from many fields such as statistics, information theory, computational complexity etc. the best way to learn the course is to view it from all these perspectives and understand the problem settings, algorithms and assumptions that underlie each. The primary goal of this course is to provide such an exposure to Post graduate students.

Programme 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. An ability to design a system or component, or process to meet stated specifications
d. An ability to identify, formulate and solve engineering problems
k. Graduate who can participate and succeed in competitive examinations.

Competencies

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

1. Understand how to program a computer to learn and improve automatically with experience
2. Understand the information processing algorithms for machine learning which might lead to a better understanding of human learning abilities
3. Identify the role of learning paradigms, algorithms, theoretical results and applications
4. Determine the fundamental relationship among the number of training examples observed, the number of hypotheses under consideration and the expected error in learned hypotheses
5. Apply the learning ability of computer to open up many new uses of computers and new levels of competence and customization
6. Apply the machine learning algorithms to learn to recognize spoken words, learn to drive an autonomous vehicle and learn to play world class backgammon game
7. Analyze the performance of the learning task through experience

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	40
4	Analyze	20	30	40
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1) Define: Well posed learning problem
2) List the various measures to improve the learning experience
3) What is concept learning?
4) What is a hypotheses?
5) What is a target function?

Understand

1) How does learning performance vary with number of training examples?
2) Which learning algorithm is most appropriate for various types of learning tasks?
3) Identify the appropriate hypotheses for different kinds of target functions.
4) How will you search a very large space of possible hypotheses to determine one that best fits the observed data?
5) Outline algorithms perform best for which types of problems and representations?

Apply

1) Demonstrate how to improve the performance of a task by applying machine learning algorithms?
2) Illustrate how machine learning algorithms are applied in Spoken word recognition, playing backgammon games, control an autonomous vehicle etc.
3) In what settings will particular algorithms converge to the desired function given sufficient training data?
4) Apply the learning algorithm designed for Checkers problem to Tic-Tac-Toe
5) Apply Naïve Bayes classifier to classify text documents.

Analyze

1)Analyze how prior knowledge held by the learner guides the process of generalizing from examples.

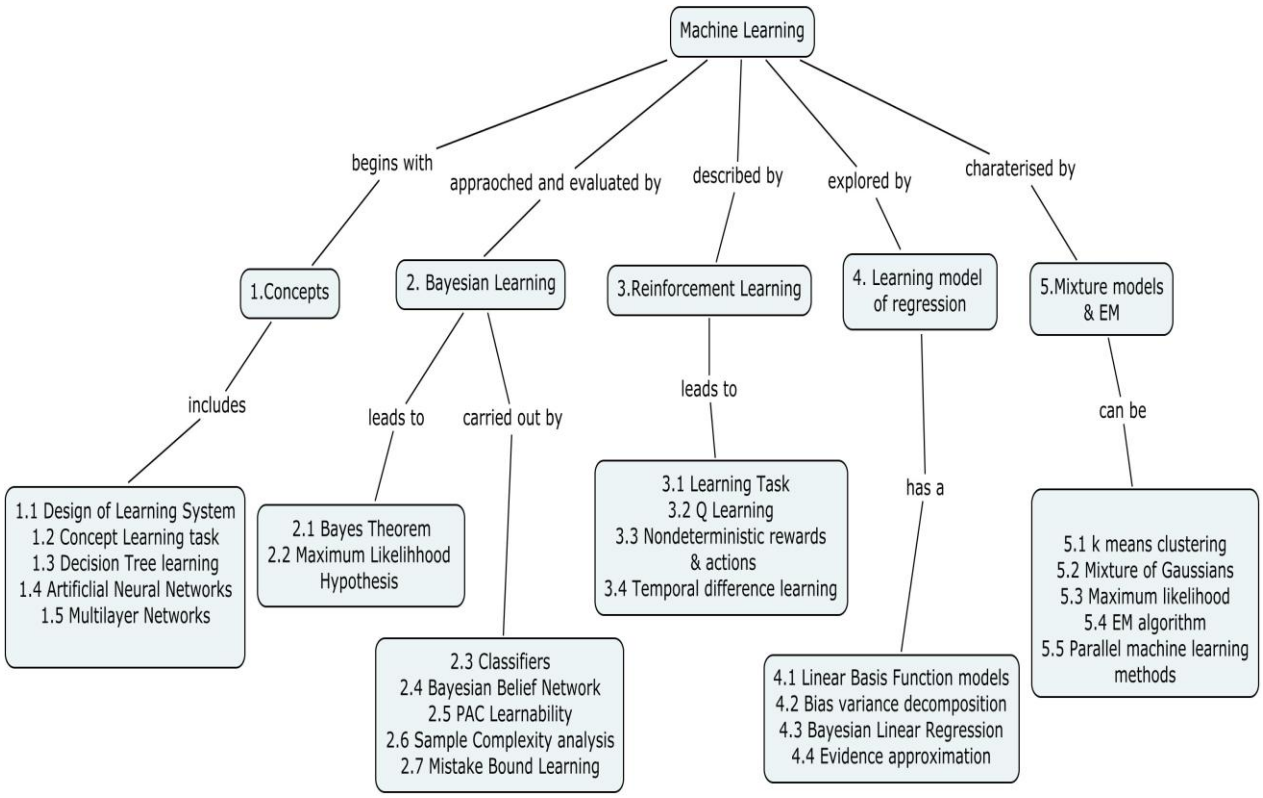
2)Analyze the best strategy for choosing a useful training experience for a learning task.

3)Analyze the specific functions the system attempts to learn and the process to be automated.

4)Analyze the general bounds that relate the confidence in learned hypotheses.

5)Analyze how a learner automatically alters the representation to improve the learning ability.

Concept Map



Syllabus

Introduction – Designing a learning system, Concept learning based on symbolic or logical representations, Decision tree learning and the problem of overfitting data, Artificial Neural networks. **Bayesian Learning –** Optimal classifier – Belief networks - Computational learning theory - PAC learning model and Mistake bound learning. **Reinforcement learning** – Q learning, Nondeterministic Rewards and Actions- Temporal difference learning. **Linear Model of Regression** – Linear basis function models – Bias variance decomposition – Bayesian Linear Regression – Evidence approximation. **Mixture models and EM –** k means clustering – Mixture of Gaussians – EM algorithm - Practical aspects of implementing Parallel machine learning methods

Reference Books

1) Tom M. Mitchell, "Machine learning", McGraw Hill,1997.
2) Ethem Alpaydin, "Introduction to machine learning", The MIT Press,2004.
3) Christopher M.Bishop, "Pattern recognition and machine learning", Springer,2007.
4) Stephen Marsland, "Machine learning: An algorithmic perspective",CRC,2009

Course Content and Lecture Schedule

Sl No.	Topics	No.of Lectures
1	**Introduction to learning system(10)**	
1.1	Design of learning system	2
1.2	Concept learning task	2
1.3	Decision Tree learning	2
1.4	Artificial Neural networks	2
1.5	Multilayer networks	2
2	**Bayesian Learning (11)**	
2.1	Bayes Theorem	1
2.2	Maximum Likelihood hypothesis	1
2.3	Classifiers	2
2.4	Bayesian Belief networks	1
2.5	PAC learnability	2
2.6	Sample complexity analysis	2
2.7	Mistake bound learning	2
3	**Reinforcement Learning(10)**	
3.1	Learning Task	2
3.2	Q Learning	3
3.3	Nondeterministic rewards and actions	3
3.4	Temporal difference learning	2
4	**Learning model of regression (9)**	
4.1	Linear Basis Function Models	3
4.2	Bias variance decomposition	2
4.3	Bayesian Linear Regression	2
4.4	Evidence approximation	2
5	**Mixture models and EM (10)**	
5.1	k means clustering	2
5.2	Mixture of Gaussians	1

5.3	Maximum likelihood	2
5.4	EM algorithm	3
5.5	Practical aspects of implementing Parallel machine learning methods	2
	Total Periods	50

Course Designer

1.Dr.S.Mercy Shalinie (shalinie@tce.edu)

Sub Code	Lectures	Tutorial	Practical	Credit
CTD	4	0	-	4

CTD Wireless Ad Hoc Networks 4:0

Preamble The course aims at exploring the concepts of wireless networks, protocols, architectures and applications.

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. An understanding of the basic principles of wireless networking and their standards.
2. An understanding of wireless network topologies and ability to perform a network planning.
3. An understanding of wireless network operations covering mobility and power management.

Assessment Pattern

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

Course Level Learning Objectives

Remember

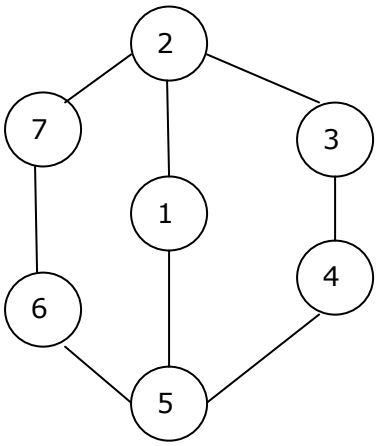
1. List the three important radio propagation phenomena at high frequencies.
2. Identify and list the limitation of Table driven routing protocols.
3. State the advantages of TORA.
4. Define confidentiality.
5. State the different classifications of energy management schemes.

Understand

1. Summarize the different issues in Ad hoc wireless networks.
2. Explain the RTS –CTS Mechanism involved in CSMA/CA.
3. Explain the concept of location aided routing.
4. Explain the working of source initiated protocols.
5. Explain the concept of security aware AODV protocol.

Apply

1. Assume that when the current size of congestion window is 48 KB, the TCP sender experiences a timeout. What will be the congestion window size if the next three transmission bursts are successful? Assume that MSS is 1 KB. Consider TCP tahoe and TCP Reno.
2. For the network shown in figure construct the fisheye routing table for nodes 7 and 5.



3. Calculate the probability of a path break for an eight-hop path, given that the probability of line break is 0.2.
4. In a military vehicular ad hoc wireless network using PRTMAC, formed by 500 nodes distributed uniformly in a battlefield area of 1000 m x 1000 m, calculate the number of nodes contending for the data channel and for control channel. The transmission range of data channel is 250 m.
5. In a military vehicular ad hoc wireless network using PRTMAC, formed by 500 nodes distributed uniformly in a battlefield area of 1000 m x 1000 m, calculate the number of nodes contending for the data channel and for control channel. The transmission range of data channel is 250 m. Also find the probability that a beacon gets collided, when the beacons are generated periodically with a period of 10 seconds. Assume the beacon length to be equal to 1 ms.

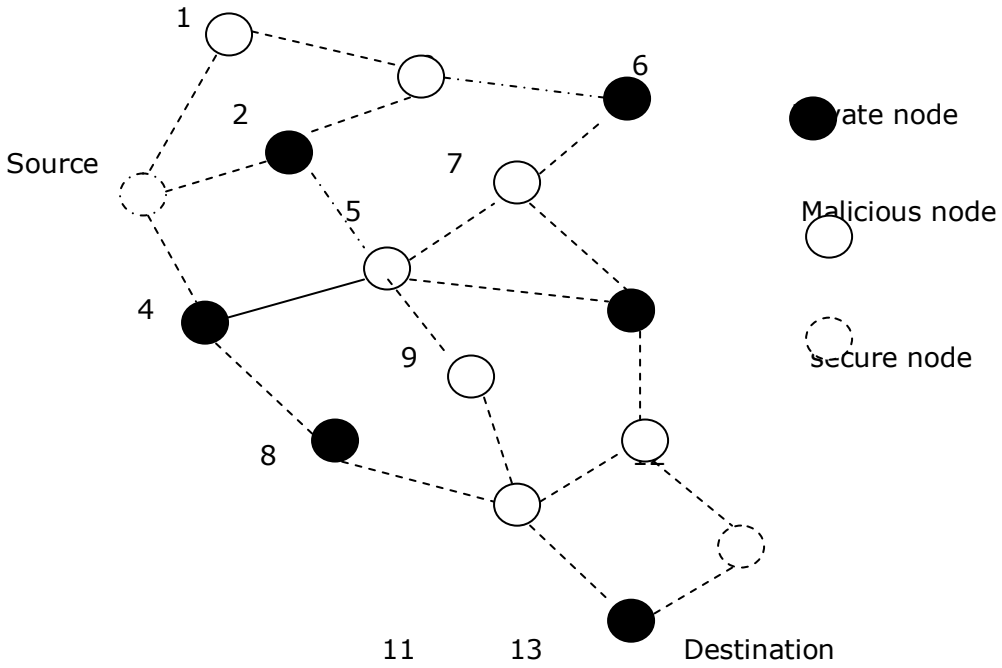
Analyze

1. Compare the different TCP solutions for Ad hoc wireless networks.
2. Examine the different phases of Associativity- Based Ad hoc Multicast routing.
3. Compare the various secure routing methods used in Ad hoc networks.
4. Examine the system power management schemes.
5. With A neat block diagram explain the concept of INSIGNIA Qos framework.

6. Compare topology reorganization in DSDV and CGSR routing protocols.

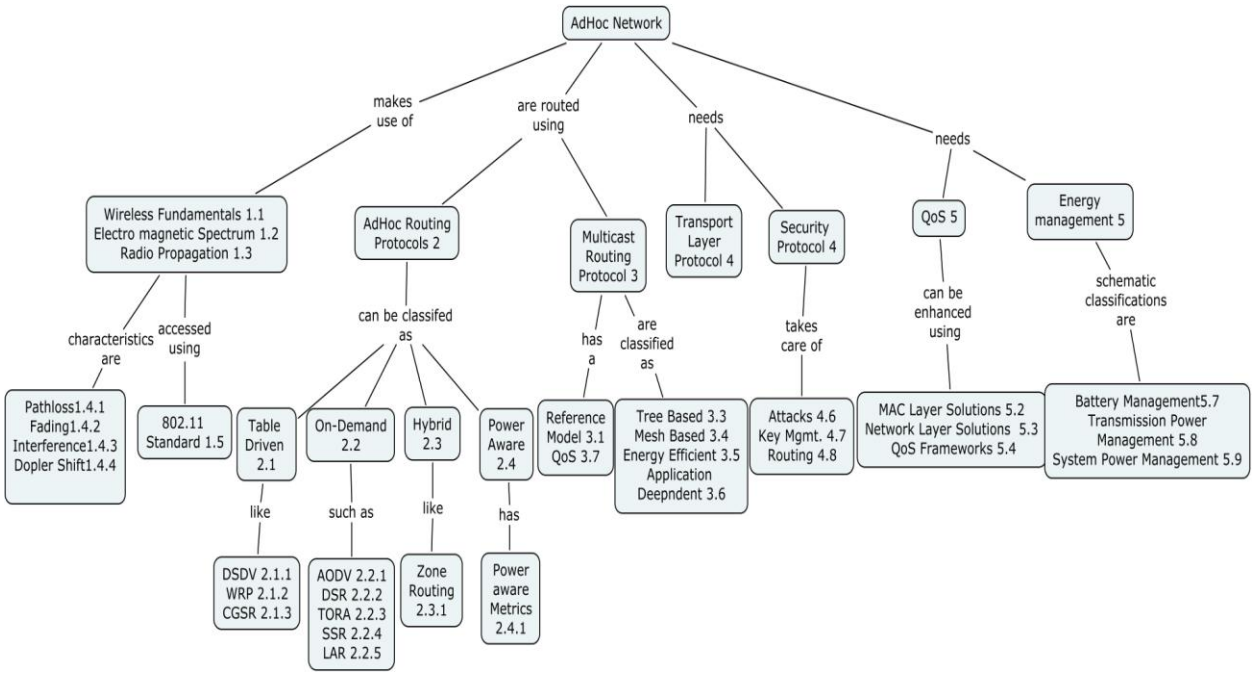
Evaluate

1. Nodes A and B want to establish a secure communication, and node A generates a random key 11001001. Suppose the functions used by both the nodes A and B for encryption is XOR, and let node A generate a random transport key 10010101 and let node B generate 00101011.Sketch the three pass Shamir protocol exchanges.
2. Predict the possible steps of the algorithms executed at the source and the intermediate nodes of an ad hoc wireless network that follow the following strategies: a) random energy b) pay-for-it strategy. Assume a session between source s and destination d. let R(s,d) be the set containing available routes between s and d , sympathy(k,r) be the the kth node in route r, and credit(k,r) and debit(k,r) be the credit and debit of kth node in route r respectively.
3. Mark the paths chosen by the following secure-routing protocols for the network topology shown in figure: a) Shortest path routing and b) SAR protocol. Assume that node 2 is secure node. C) If node 2 (which lies in the path chosen by SAR protocol) is suddenly attacked and becomes a malicious node, then mark an alternative path chosen by SAODV protocol.



4. Estimate the approximate control overhead for the ODMRP protocol over a 200 second time period. Assume that all nodes are stationary. Number of nodes: 50 . Time period for sending a JoinReq : 2 secs.
5. Estimate the approximate control overhead for the DCMP protocol over a 200 second time period. Assume that all nodes are stationary.S1 is a active source and S2 is a passive source. Number of nodes: 50 . Time period for sending a JoinReq : 2 secs.

Concept Map



Syllabus

Introduction :Introduction-Fundamentals of Wireless Communication Technology - The Electromagnetic Spectrum - Radio Propagation Mechanisms - Characteristics of the Wireless Channel - IEEE 802.11 Standard –**Ad Hoc Routing Protocols**: Issues and Challenges - Classifications of Routing Protocols-Table-Driven Routing Protocols - On-Demand Routing Protocols – Hybrid Routing Protocols- Power-Aware Routing (PAR) - **Multicast routing In Ad Hoc Networks**: - An Architecture Reference Model for Multicast Routing Protocols -Classifications of Multicast Routing Protocols- Tree-Based Multicast Routing Protocols- Mesh-Based Multicast Routing Protocols-Energy-Efficient Multicasting - Multicasting with Quality of Service Guarantees – Application-Dependent Multicast Routing **Transport Layer, Security Protocols**: Designing a Transport Layer Protocol - Design Goals of a Transport Layer Protocol - Classification of Transport Layer Solutions - TCPOver Ad Hoc Wireless Networks -Other Transport Layer Protocols - Security Requirements - Issues and Challenges in Security Provisioning -Network Security Attacks - Key Management - Secure Routing **Qos and Energy Management**: Classifications of QoS Solutions - MAC Layer Solutions-Network Layer Solutions - QoS Frameworks for Ad Hoc Wireless Networks Energy Management in Ad Hoc Wireless Networks –Introduction - Need for Energy Management in Ad Hoc Wireless Networks - Classification of Energy Management Schemes - Battery Management Schemes - Transmission Power Management Schemes - System Power Management Schemes

References

1.C.Siva Ram Murthy and B.S. Manoj "Ad Hoc Wireless Networks: Architectures and Protocols", Pearson education, 2008

2.Charles E. Perkins, Ad Hoc Networking, Addison Wesley, 2000

3.William Stallings, "Wireless Communications and Networks", Pearson education, 2003

4.J. Schiller, "Mobile Communications", Pearson education, 2003

5.Vijay K. Garg, "Wireless Communications and Networking", Elsevier, 2008.

Course Contents and Lectures schedule

1	**AdHoc Networks Introduction (8)**	
1.1	Fundamentals of Wireless Communication	2
1.2	The Electromagnetic Spectrum	1
1.3	Radio Propagation Mechanisms	2
1.4	Characteristics of the Wireless Channel	1
1.5	IEEE 802.11 Standard	2
2	**Ad Hoc Routing Protocols (12)**	
2.0	Issues and Challenges	2
2.1	Table-Driven Routing Protocols	3
2.2	On Demand Routing Protocols	3
2.3	Hybrid Routing Protocols	2
2.4	Power-Aware Routing	2
3	**Multicast routing In Ad Hoc Networks (10)**	
3.1	An Architecture Reference Model for Multicast Routing Protocols	**2**
3.2	Classifications of Multicast Routing Protocols	1
3.3	Tree-Based Multicast Routing Protocols	2
3.4	Mesh-Based Multicast Routing Protocols	2
3.5	Energy Efficient Routing Protocols	1
3.6	Application Dependent Protocols	1
3.7	Multicasting with QoS Guarantee	1
4	**Transport Layer, Security Protocols (10)**	
4.1	Designing a Transport Layer Protocol	2

4.2	Design Goals of a Transport Layer Protocol	1
4.3	TCP Over Ad Hoc Wireless Networks	2
4.4	Other Transport Layer Protocols	1
4.5	Security Requirements - Issues and Challenges in Security Provisioning	1
4.6	Network Security Attacks	1
4.7	Key Management	1
4.8	Secure Routing	1
5	**Qos and Energy Management (10)**	
5.1	Classifications of QoS Solutions	1
5.2	MAC Layer Solutions	1
5.3	Network Layer Solutions	1
5.4	QoS Frameworks	1
5.5	Energy Management	1
5.6	Classification of Energy Management Schemes -	1
5.7	Battery Management Schemes -	2
5.8	Transmission Power Management Scheme	1
5.9	System Power Management Schemes	1
	Total number of hours	**50**

Course Designer

1. C.Senthilkumar cskcse@tce.edu
2. S.Prasanna sprcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTE	4	-	-	4

CTE Distributed Operating Systems Concepts and Principles 4:0

Preamble: This course is offered in the Second semester for the students of Post Graduate Computer Science and Engineering. One of the fundamental building blocks of a Computer Science course is the Operating System Concepts. The distributed OS Concepts dealt in this subject are common to most of the operating systems available today. The case studies and illustrations help in understanding the concepts clearly. Concise definition of concepts and examples adds flavor to this subject.

Program Outcomes addressed

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

k. Graduates will demonstrate an ability to use techniques, skills and modern engineering tools necessary for engineering practice

d. Graduates will demonstrate an ability to function in multi-disciplinary terms

Competencies

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

1. Introduce the basic concepts of Distributed Operating System
2. Discuss in detail about the Distributed Shared Memory
3. Express the Distributed File systems models with a case study
4. Understand about the Fault Tolerance

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	20	20	20	20
2	Understand	40	40	40	40
3	Apply	40	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. Define Network OS and Distributed OS
2. What is Multicomputer and Multiprocessor?
3. Name the three important features used to differentiate the distributed OS and Network OS
4. What are the fields in RPC Call Message Format?
5. What are the issues related to Synchronization?
6. What is drifting of clocks?
7. State the requirements to be satisfied by an algorithm for implementing mutual exclusion
8. Write the sequence of events required to use a resource by a process ?
9. State necessary conditions for deadlocks
10. What are failure models?

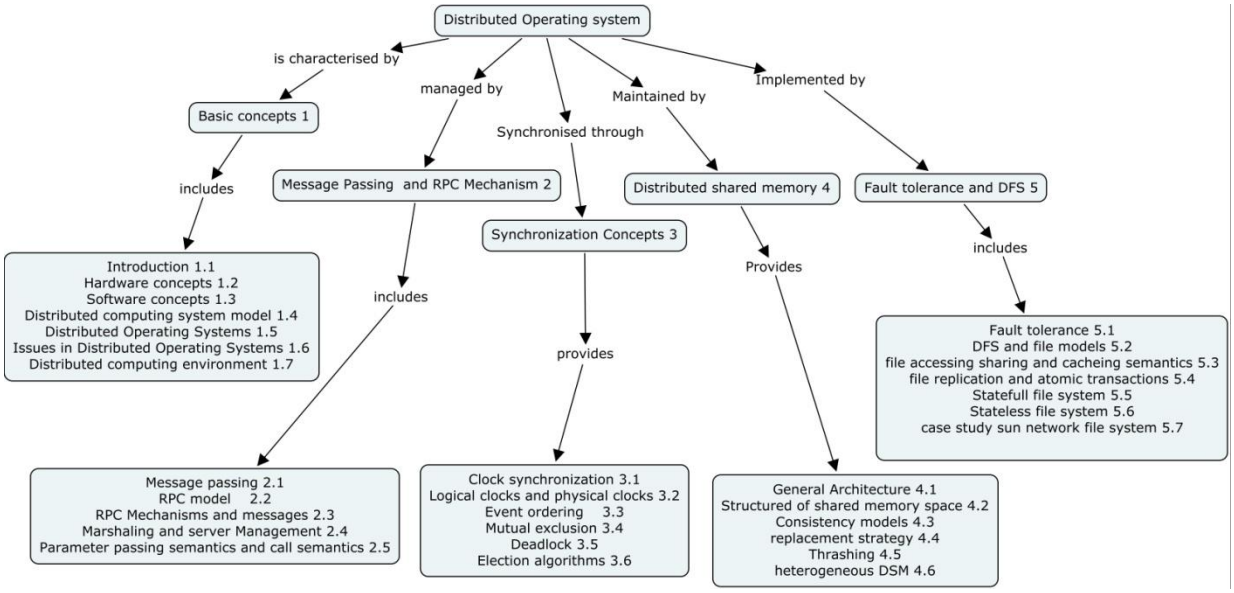
Understand

1. Discuss about the advantages and disadvantages of commonly used models in configuring Distributed Operating Systems.
2. Why are Distributed Computing Systems gaining popularity ?
3. Describe in detail about DCE
4. Explain the general design issues of Servers
5. Discuss about the RPC Model
6. Discuss about the RPC Messages
7. Discuss about the Marshaling arguments and results
8. How the computer clocks are implemented?
9. Discuss about the necessary and sufficient conditions of deadlocks
10. How to handle and prevent deadlocks in distributed systems

Apply

1. Demonstrate the Distributed deadlock detection algorithms
2. Implement RPC Mechanism
3. Apply the server implementation to stateful and stateless servers
4. Illustrate the method of event ordering in Distributed Systems
5. Model a distributed deadlock using graphical method
6. Construct a resource allocation graph.
7. Illustrate Election algorithms in Distributed Systems
8. Demonstrate the structure of the shares memory space.
9. Illustrate the principles of Distributed File Systems with SUN NFS
10. Discuss about the Distributed File System Models

Concept Map



Syllabus

Distributed Systems - Introduction, Hardware concepts software concepts – Distributed Computing System Models – Distributed Operating Systems – Issues in Distributed Operating Systems – Distributed Computing Environment **Message passing and RPC Mechanism** – Message passing – Remote Procedure call model – Implementing RPC mechanism – RPC Messages – Marshaling – Server Management Parameter passing semantics – call semantics **Synchronization** – Clock Synchronization – logical clocks – physical clocks – event ordering – mutual exclusion – deadlock – election algorithms – **Distributed Shared Memory** – General Architecture – structure of shared memory space – consistency models – replacement strategy – thrashing – heterogeneous Distributed Shared Memory. **Fault Tolerance and Distributed File Systems -** Fault tolerance – Distributed File Systems **–** File models – file accessing models – file sharing semantics – file caching semantics – file replication – atomic transactions – design principles – Stateful and Stateless File Systems - case studies –Network File Systems

References:

1. Pradeep K. Sinha, Distributed Operating Systems Concepts and Design, Prentice Hall of India Private Limited, 2008

2. Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems Principles and Paradigms, Pearson Education, 2007

D M Dhamdhere, Operating Systems A Concept Based Approach: Tata McGraw-Hill Publishers, 2003, Second Reprint.

Course contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Distributed System**	
1.1	Introduction of Distributed Systems	2
1.2	Hardware Concepts	2
1.3	Software Concepts	1
1.4	Distributed Computing System Models	1
1.5	Distributed Operating Systems	1
1.6	Issues in Distributed Operating Systems	2
1.7	Distributed Computing Environment	1
2	**Message Passing and RPC Mechanism**	
2.1	Message passing	2
2.2	Remote Procedure call model	2
2.3	Implementing RPC mechanism and RPC Messages	2
2.4	Marshaling and Server Management	2
2.5	Parameter passing semantics and call semantics	2
3	**Synchronization**	
3.1	Clock Synchronization	1
3.2	logical clocks and physical clocks	2
3.3	Event ordering	1
3.4	Mutual exclusion	2
3.5	Deadlock	3
3.6	Election algorithms	1
4	**Distributed Shared Memory**	
4.1	General Architecture	1
4.2	structure of shared memory space	2
4.3	consistency models	2
4.4	replacement strategy	2
4.5	Thrashing	2
4.6	Heterogeneous Distributed Shared Memory	1
Fault Tolerance and Distributed File Systems		
5.1	Fault Tolerance	2
5.2	Distributed File Systems and File models	2

5.3	file accessing models, file sharing semantics and file caching semantics	2
5.4	file replication and atomic transactions	1
5.5	Stateful file systems	1
5.6	Stateless file systems	1
5.7	case study - SUN Network File Systems	1
	Total	50

Course Designers:

1. Mr.R.Chellamani rcmcse@tce.edu
2. Dr.T. Subbulakshmi subbulakshmi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTF	4	-	-	4

CTF Multimedia Information Systems 4:0

Preamble

The goal of this subject is to provide comprehensive coverage of principles of multimedia elements, various compression techniques, including leading algorithms for various applications. This course will motivate the students to acquire knowledge in current technology and tools of multimedia for various applications.

Programme Outcomes addressed

a. Graduates will demonstrate knowledge of mathematics, science and engineering.
b. An ability to identify, formulate and solve engineering problems.
d. Graduates who can participate and succeed in competitive examinations.

Competencies

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

1. Explain the elements and various File formats of multimedia.
2. Discuss the need and benefits of Multimedia.
3. Select Image, Audio and Video editing tools for different applications.
4. Explain and compare algorithms for text, image, audio and video compressions.
5. Explain the differences between ordinary file system and multimedia file system.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	30	30	20
2	Understand	20	20	10
3	Apply	50	50	40
4	Analyze	0	0	30
5	Evaluate	0	0	0
6	Create	0	0	0

Learning Objectives

Remember

1. What is meant by the terms Multimedia and Hypermedia?.
2. List out the color models.
3. Where CMYK color model is used widely?
4. What are the components of JPEG?
5. Write any four applications of multimedia.

6. Explain data and file format standards.
7. Explain the different properties of H.261 and H.263?
8. State and explain the characteristics of multimedia DBMS.
9. Explain the phases of Resource Reservation and Management Process.

Understand

1. What is the difference between static and dynamic media?. Give two examples of each type.
2. Differentiate Lossless from Lossy Compression.
3. Why data compression is highly desirable for multimedia activities?
4. Compare and Contrast different MPEG Standards.
5. Differentiate between the BMP, JPG, GIF and TIFF image file formats mentioning where each is used?
6. Differentiate between Image editing and Animation tools.
7. How multimedia database is differed from Conventional database?.
8. With the aid of the diagram, describe the interlaced mode of operation of GIF.

Apply

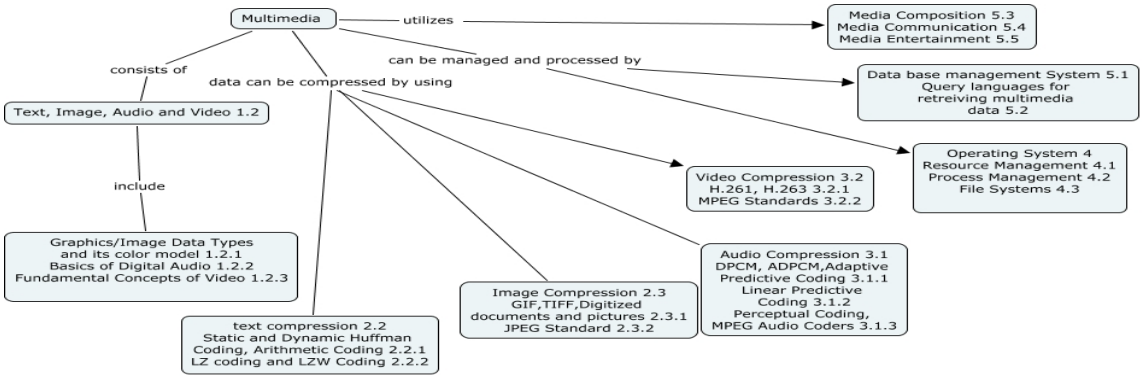
1. Show how will you use Huffman coding to encode the following set or tokens:
   BABACACADADABBCBABEBEDDABEEEBB
   a. How is the message transmitted when encoded?.
   b. How many bits are needed to transfer this coded message?
2. An MPEG-1 video has a frame sequence : IBBPBBPBBPBBI. Determine the size of GOP. Derive the transmission sequence of the frames?.
3. A 15 inch monitor with an aspect ratio of 4:3 has a pixel addressability of 800 × 600. Calculate its resolution?
4. Show all the steps of LZW compression including dictionary formation for the text string "ABABAABA$ABAAC$AB"
5. Assuming a quantization threshold value of 16, find the resulting quantization error for each of the DCT Co-efficients
   127, 72, 64, 56,-56,-64,-72,-128
6. A digitized video is to be compressed using MPEG-I standard. The frame sequence is given by IBBPBBPBBPBBI. And average compression ratios of 10:1 (I), 20:1 (P), and 50:1 (B) , calculate the average bit rate that is generated by the encoder for both the NTSC and PAL digitations formats.
7. A series of messages is to be transferred between two computers over a PSTN. The messages comprise just the characters A through H. Analysis has shown that the probability(Relative frequency of occurrence) of each charatcet is as follows:
   A and B= 0.25 C and D= 0.14, E, F G and H= 0.055
   a) Use Shannon's formula to derive the minimum average number of bits per character.

b) Use Huffman Coding to derive a codeword set and prove this is the minimum set by constructing the corresponding Huffman code tree.

Analyze

1. Compare and contrast static Huffman coding with dynamic Huffman coding and hence analyze which algorithm is efficient?
2. What are the advantages of MPEG standards over other video compression algorithms?
3. Analyze the performance of Query languages for retrieving Multimedia Data with SQL.
4. Analyze the factors that influence the running time of a multimedia product.
5. What are the advantages of JPEG standard over other image compression algorithms?

Concept Map



Syllabus

Introduction to Multimedia: Overview of Multimedia, Multimedia – Text, Image, Audio, and Video File formats, Graphics / Image Data types, Color models in Images. Basics of Digital Audio: MIDI- Overview, Hardware aspects of MIDI, Structure of MIDI messages, General MIDI and MIDI to WAV Conversion, Quantization and transmission of audio- Coding of audio, PCM, Differential coding of Audio. Fundamental Concepts in Video: Types of Video Signals, Analog Video, Digital Video. **Text and Image Compression**: Compression Principles- Types of Compression, Entropy Encoding, Source Encoding. Text Compression- Static Huffman Coding, Dynamic Huffman Coding, Arithmetic Coding, LZ Coding and LZW Coding. Image Compression- GIF,TIFF, Digitized documents, Digitized pictures and JPEG Standard.**Audio and Video Compression:** Audio Compression- DPCM, ADPCM, Adaptive Predictive Coding, Linear Predictive Coding, Perceptual Coding, MPEG Audio coders. Video Compression- Principles, H.261, H.263, MPEG Standards. **Multimedia Operating System:** Real Time, Resource Management-Resources, Requirements, Components and Phases, Allocation scheme, Continuous Media Resource Model. Process Management-Real time process Management, requirements, Traditional real-time scheduling, System model, Earliest Deadline and Rate Monotonic Algorithms, File systems, Additional OS Issues. **Multimedia Databases and Applications of Multimedia:** Multimedia Databases: Design and Architecture of Multimedia Database, Query languages

for retrieving Multimedia Data. Applications of Multimedia: Media Composition-Text and Graphics Editors, Image and Animation Editors, Audio and Video Editors, Media Communication, Media Entertainment – Virtual Reality, Interactive Audio and Video. Content Based Image Retrieval

References

1. Fred Halsal, "Multimedia communications", Pearson Education, 2001.
2. Ralf Steinmetz and Klara Nahrstedt , " Multimedia Computing, Communications and Applications", Pearson Education, 2004.

3.Ze-Nian Li, Mark S.Drew, "Fundamentals of Multimedia", PHI, 2006.

4.S.Subramanian, "Multimedia Database Systems", Morgan Kaufmann Publishers, 2001.

5.Mark Nelson , " The Data Compression Book", BPB Publications, 1993.

6.Khalid Sayood, " Introduction to Data Compression", Academic Press, 2000.

7.B.prabhakaran, "Multimedia Database Management Systems", Springer International Edition,2007.

Course contents and Lecture Schedule

No	Topic	No of Lectures
1.	**INTRODUCTION TO MULTIMEDIA (7)**	
1.1	Overview of Multimedia	1
1.2	Multimedia – Text, Image, Audio, and Video File formats	2
1.2.1	Graphics / Image Data types, Color models in Images	1
1.2.2	Basics of Digital Audio: MIDI- Overview, Hardware aspects of MIDI, Structure of MIDI messages, General MIDI and MIDI to WAV Conversion, Quantization and transmission of audio- Coding of audio, PCM, Differential coding of Audio	2
1.2.3	Fundamental Concepts in Video: Types of Video Signals, Analog Video, Digital Video.	1
2	**TEXT AND IMAGE COMPRESSION (12)**	
2.1	Compression Principles- Types of Compression, Entropy Encoding, Source Encoding	2
2.2	Text Compression	1
2.2.1	Static Huffman Coding, Dynamic Huffman Coding, Arithmetic Coding,	3
2.2.2	LZ Coding and LZW Coding	2
2.3	Image Compression	1
2.3.1	GIF,TIFF ,Digitized documents, Digitized pictures	1
2.3.2	JPEG	2

3.	**Audio and Video Compression (12)**	
3.1	Audio Compression	1
3.1.1	DPCM, ADPCM, Adaptive Predictive Coding	1
3.1.2	Linear Predictive Coding	1
3.1.3	Perceptual Coding, MPEG Audio coders	2
3.2	Video Compression- Principles	2
3.2.1	H.261, H.263	2
3.2.2	MPEG standards	3
4	**Multimedia Operating System (10)**	
4.1	Real Time, Resource Management-Resources, Requirements, Components and Phases, Allocation scheme, Continuous Media Resource Model	4
4.2	Process Management-Real time process Management, requirements, Traditional real-time scheduling, System model, Earliest Deadline and Rate Monotonic Algorithms	2
4.3	File systems, Additional OS Issues.	4
5	**Multimedia Databases and Applications of Multimedia (9)**	
5.1	Multimedia Databases: Design and Architecture of Multimedia Database	1
5.2	Query languages for retrieving Multimedia Data	3
5.3	Applications of Multimedia: Media Composition-Text and Graphics Editors, Image and Animation Editors, Audio and Video Editors	2
5.4	Media Communication	2
5.5	Media Entertainment – Virtual Reality, Interactive Audio and Video, Content Based Image Retrieval	1
	Total Number of hours	**50**

Course Designer:

S. Sridevi sridevi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTG	4	-	-	4

CTG Intelligent Optimization Algorithms 4:0

Preamble

Intelligent optimization Algorithms investigate several techniques commonly referred to as intelligent algorithms, and takes a pragmatic engineering approach to the design, evaluation, and implementation of intelligent systems. The algorithmic family includes genetic algorithms, hill-climbing, simulated annealing, ant colony optimization, particle swarm optimization, and so on. The course will enable the students to familiarize the various concepts of these intelligent optimization techniques, such as what they are, how they work, how to design, build and implement them.

Program Outcomes addressed

a. Graduates will able to apply knowledge of engineering, information technology, mathematics and science

c. Graduates will design a system or component or process to meet stated specifications

d. Graduates will identify, formulate and solve engineering problems

g. Graduates will demonstrate their ability to function on multidisciplinary teams

Competencies

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

1. Explain the difference between Single-state Methods and Population Methods?
2. Identify the suitable solution representation for the given problem?
3. Explain various methods to parallelize the given problem?
4. Identify different objectives in the given problem and apply suitable multi-objective optimization algorithm?
5. Identify an optimized model to classify the given instances?

Assessment Pattern

S.N	Bloom's Category	Test 1	Test 2	Test 3/End-Semester examination
1	Remember	30	20	20
2	Understand	30	60	30
3	Apply	40	20	30
4	Analyze	0	0	20
5	Evaluate	0	0	0
6	Create	0	0	0

Course Level Learning Objectives

Remember

1. Define Random search for Hill Climbing Algorithm?
2. Write the basics steps involved in metaheuristics algorithms to optimize the candidate solution?
3. Write the difference between unimodal and multimodal optimization problem?
4. What are the parameters used in PSO for exploration and exploitation?
5. What is overshooting in Gradient Ascent Algorithm?
6. What are the ways to parallelize a problem

Understand

1. Explain the Feature-based Tabu Search algorithm?
2. Enumerate and explain the selection mechanisms in GA?
3. Explain Tree encoding and Cellular encoding?
4. Explain the methods used to maintain diversity in population?
5. How Pareto strength is maintained in Multi-objective problem?
6. Explain Univariate Estimation of Distribution Algorithms for Model fitting?

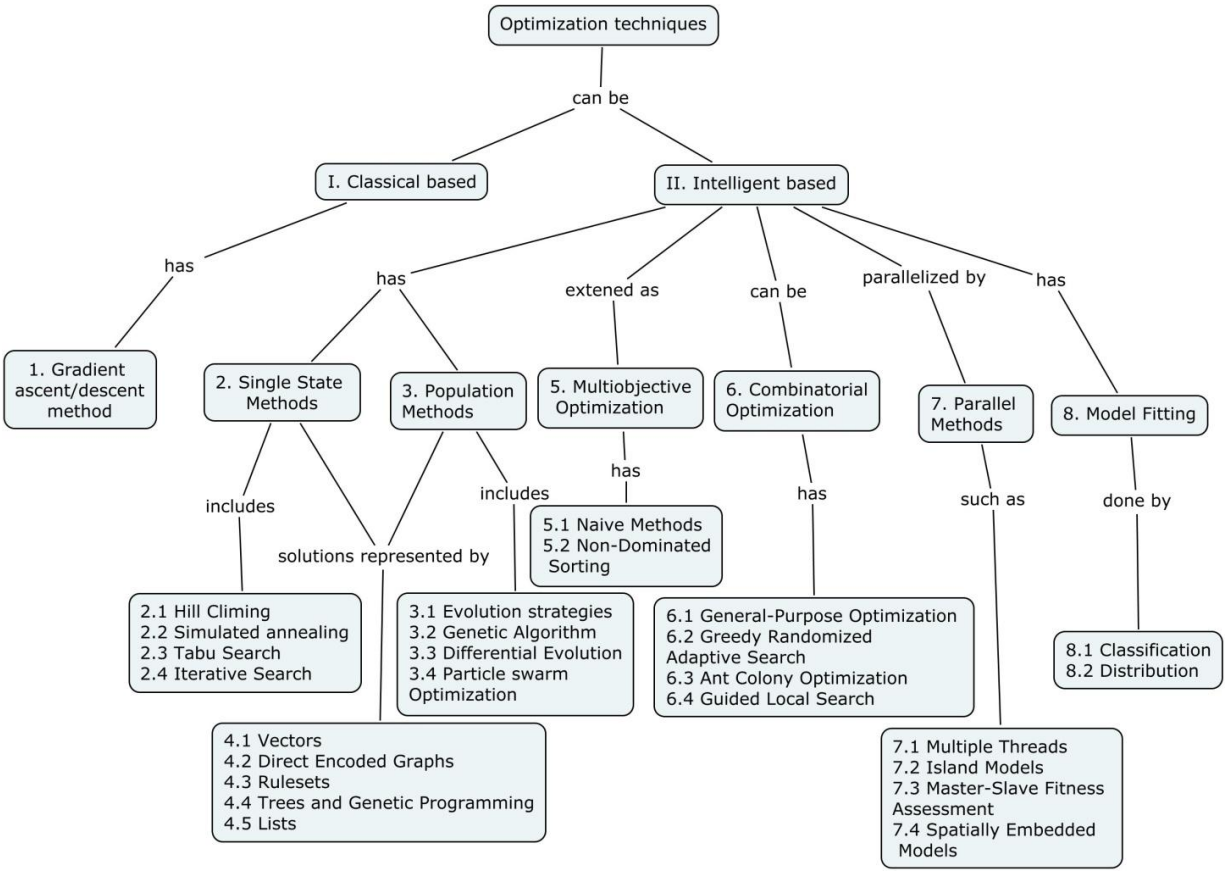
Apply

1. Construct Tabu Search algorithm for Classical Vehicle Routing Problem?
2. How Scatter Search algorithm can be applied to Path Relinking problem?
3. Construct an effective ACO algorithm for DNA sequencing problem?
4. Construct an effective PSO algorithm for Maximum Clique problem?
5. Construct GA for vertex coloring problem?
6. Identify a suitable parallelizing model to parallelize TSP using DE?

Analyze

1. Examine the performance of RGA with PSO for the simple test bed problem?
2. Compare the performance of popular meta-heuristic algorithms based on complex parameter interactions for TSP?
3. Examine the performance of parallel GA for TSP using Island models and Master-Slave models.

Concept Map



Syllabus Classical Optimization Techniques – Gradient Ascent/Descent Methods- Drawback of classical techniques – Need for Intelligent techniques- Single-State Methods – Hill Climbing - Simulated Annealing - Tabu Search -Iterated Local Search **Population Methods** - Evolution Strategies- The Genetic Algorithm- Differential Evolution - Particle Swarm Optimization **Representation** – Vectors - Direct Encoded Graphs – Rulesets - Trees and Genetic Programming - Lists, Machine-Language Genetic Programming, and Grammatical Evolution **Multiobjective Optimization** -Naive Methods - Non-Dominated Sorting - Combinatorial Optimization -General-Purpose Optimization and Hard Constraints- Greedy Randomized Adaptive Search Procedures - Ant Colony Optimization -Guided Local Search **Parallel Methods** - Multiple Threads- Island Models - Master-Slave Fitness Assessment - Spatially Embedded Models - Model Fitting - Model Fitting by Classification - Model Fitting with Distribution

References

1. Michel Gendreau, Jean-Yves Potvin, "Handbook of Metaheuristics", Second Edition, International Series in Operations Research & Management Science ,Springer 2010.

2. Godfrey C. Onwubolu, B. V. Babu, "New optimization techniques in engineering", Series: Studies in Fuzziness and Soft Computing, Vol. 141, Springer, 2004.

3. Sean Luke , "Essentials of Metaheuristics", (Online Version 1.2), lulu.com (March 5, 2011).

4. Crina Grosan, Ajith Abraham, "Intelligent Systems: A Modern Approach", Intelligent Systems Reference Library,Volume 17, Springer, 2011.

5. D.T. Pham , D. Karaboga, "Intelligent Optimisation Techniques", Springer, 2000.

Course Contents and Lectures schedule

S.No	Topics	No of Lectures
I	**Classical Optimization Techniques**	
1	Gradient Ascent/Descent Methods– Drawback of classical techniques- Need for Intelligent techniques	3
II	**Intelligent Optimization Techniques**	
2	**Single-State Methods**	
2.1	Hill Climbing	1
2.2	Simulated Annealing	2
2.3	Tabu Search	2
2.4	Iterated Local Search	2
3	**Population Methods**	
3.1	Evolution Strategies	2
3.2	Genetic Algorithm	3
3.3	Differential Evolution	2
3.4	Particle Swarm Optimization	3
4	**Representation**	
4.1	Vectors	2
4.2	Direct Encoded Graphs	1
4.3	Rulesets	2
4.4	Trees and Genetic Programming	1
4.5	Lists, Machine-Language Genetic Programming, and Grammatical Evolution	2

5	**Multiobjective Optimization**	
5.1	Naive Methods	1
5.2	Non-Dominated Sorting	2
6	**Combinatorial Optimization**	
6.1	General-Purpose Optimization and Hard Constraints	1
6.2	Greedy Randomized Adaptive Search Procedures	2
6.3	Ant Colony Optimization	2
6.4	Guided Local Search	2
7	**Parallel Methods**	
7.1	Multiple Threads	2
7.2	Island Models	2
7.3	Master-Slave Fitness Assessment	2
7.4	Spatially Embedded Models	2
8	**Model Fitting**	
8.1	Model Fitting by Classification	2
8.2	Model Fitting with Distribution	2
	Total	50

Course Designer

S. Sudha ssj@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

M.E DEGREE (Computer Science and Engineering) PROGRAMME

THIRD AND FOURTH SEMESTER

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2011-2012 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

Department of Computer Science and Engineering

Graduating Students of ME 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 the M.E. CSE (for those who are admitted from the year 2011-2012 onwards)

4th (12)							Project Phase II 0:12
3rd (16)	CT31 Data Warehousing and Mining Techniques 4:0	CT3Y Elective 5 4:0	CT3Y Elective 6 4:0				Project Phase I 0:4
2nd (24)	CT21 Modeling and Simulation 3:1	CT22 Software Engineering Theory and Practice 3:0	CT2Y Elective 1 4:0	CT2Y Elective 2 4:0	CT2Y Elective 3 4:0	CT2Y Elective 4 4:0	Seminar 0:1
1st (24)	CT11 Computer Architecture 4:0	CT12 Data Structures and Algorithms 3:1	CT13 Object Oriented Analysis and Design 3: 0	CT14 Network Technology 3:1	CT15 Compiler Design 3:1	CT16 Graphs and Combinatorial Algorithms 3:1	CT17 Analysis and Design Lab 0:1

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

(An Autonomous Institution affiliated to Anna University)

CURRICULAM

(For the Students admitted from the academic year 2011-2012)

Name of the Degree: M.E (Computer Science and Engineering) Programme

SUBJECTS OF STUDY

I SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT11	Computer Architecture	4	0	0	4
CT12	Data Structures and Algorithms	3	1	0	4
CT13	Object Oriented Analysis and Design	3	0	0	3
CT14	Network Technology	3	1	0	4
CT15	Compiler Design	3	1	0	4
CT16	Graphs and Combinatorial Algorithms	3	1	0	4

Practical:

CT17	Analysis and Design Laboratory	0	0	3	1

Total Credits 24

II SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT21	Modeling and Simulation	3	1	0	4
CT22	Software Engineering Theory and Practice	3	0	0	3
CT2Y	Elective 1	4	0	0	4
CT2Y	Elective 2	4	0	0	4
CT2Y	Elective 3	4	0	0	4
CT2Y	Elective 4	4	0	0	4

Practical:

CT27	Seminar	0	0	3	1

Total Credits 24

III SEMESTER

Theory:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT31	Data Warehousing and Mining Techniques	4	0	0	4
CT3Y	Elective 5	4	0	0	4
CT3Y	Elective 6	4	0	0	4

Practical:

CT34	Project	0	0	12	4

Total Credits 16

IV SEMESTER

Practical:

Sub Code	Name of the subject	Regulation			
		L	T	P	C
CT41	Project	0	0	36	12

Total Credits 12

Total No. of credits to be earned for the award of degree: 76

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E Degree (Computer Science and Engineering) Programme
SCHEME OF EXAMINATIONS
(For the candidates admitted from 2011-2012 onwards)

I 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	CT11	Computer Architecture	3	50	50	100	25	50
2	CT12	Data Structures and Algorithms	3	50	50	100	25	50
3	CT13	Object Oriented Analysis and Design	3	50	50	100	25	50
4	CT14	Network Technology	3	50	50	100	25	50
5	CT15	Compiler Design	3	50	50	100	25	50
6	CT16	Graphs and Combinatorial Algorithms						
PRACTICAL								
7	CT17	Analysis and Design Laboratory	3	50	50	100	25	50

II 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	CT21	Modeling and Simulation	3	50	50	100	25	50
2	CT22	Software Engineering Theory and	3	50	50	100	25	50

		Practice						
3	CT2Y	Elective 1	3	50	50	100	25	50
4	CT2Y	Elective 2	3	50	50	100	25	50
5	CT2Y	Elective 3	3	50	50	100	25	50
6	CT2Y	Elective 4	3	50	50	100	25	50
PRACTICAL								
7	CT27	Seminar	3	50	50	100	25	50

III 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	CT31	Data Warehousing and Mining Techniques	3	50	50	100	25	50
2	CT3Y	Elective 5	3	50	50	100	25	50
3	CT3Y	Elective 6	3	50	50	100	25	50
PRACTICAL								
4	CT34	Project	-	150	150	300	75	150

IV 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	CT41	Project	-	150	150	300	75	150

Sub Code	Lectures	Tutorial	Practical	Credit
CT31	4	0	-	4

CT31 Data Warehousing and Mining Techniques 4:0

Preamble: This course aims at facilitating the student to understand the concepts of data warehousing and various techniques involved in mining the data from the data warehouse.

Competencies

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

1. Understand the data warehouse concepts
2. Construct the data warehouse using various structural schemas.
3. Prepare the data required for analysis from the raw data.
4. Apply the mining techniques on various kinds of databases.
5. Apply the association techniques for identifying relationship between data.
6. Group the data based on their characteristics using classification and clustering techniques.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	30	20	30
3	Apply	30	60	45
4	Analyze	0	0	0
5	Evaluate	0	0	0
6	Create	20	0	15

Course Level Learning Objectives

Remember

1. Define dimension and fact table.
2. List out the types of OLAP operations.
3. Define Data Mining.
4. Define strong association rule?
5. How to select an attribute for classification?
6. Define cluster analysis?

Understand

1. Differentiate OLTP and OLAP.
2. Suppose a data warehouse consists of the four dimensions date, spectator, location and game and two measures count and charge, where charge is the fare that a spectator pays when watching a game on a given date. Spectator may be students, adults, or seniors, with each category having its own charge rate.
   Starting with the base cuboid [date, spectator, location, game] what specific OLAP operation should one perform in order to list the total charge paid by the student spectator at Nehru-stadium in 2008.
3. Compare and contrast the clustering and the classification.
4. Can we do data mining on the data generated by the web? Justify
5. Illustrate the significance of candidate set generation step of level wise algorithm.
6. Compare agglomerative and divisive hierarchical clustering

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. Suppose that the data mining task is 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. Suppose initially assign A1, B1 and C1 as the center of each cluster respectively. Use K-Means algorithm to show the final three clusters.

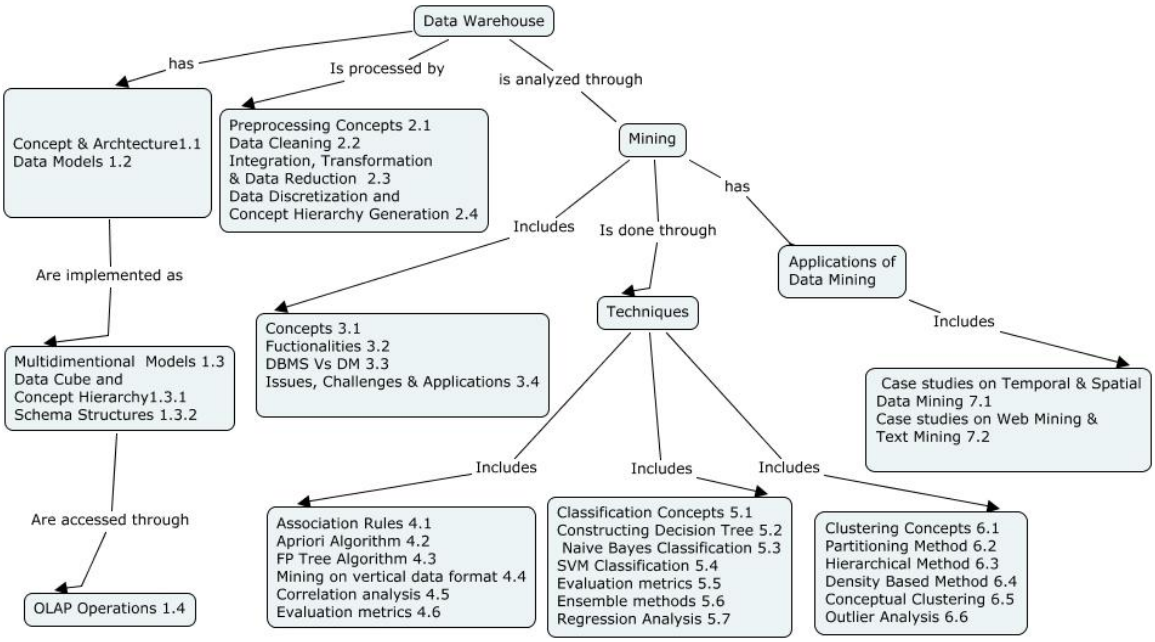
Create

1. Give the star schema for the base cuboid (student, course, semester, instructor)
2. Suppose a data warehouse consists of the four dimensions date, spectator, location and game and two measures count and charge , where charge is the fare that a spectator pays when watching a game on a given date. Spectator may be students, adults, or seniors, with each category having its own charge rate. Draw the star schema diagram for the data warehouse.
3. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit. Draw a snowflake schema diagram for the data warehouse.
4. Give the star schema for the base cuboid (employee, department, salary, attendance)
5. Suppose a data warehouse consists of the four dimensions date, spectator, location and game and two measures count and charge , where charge is the fare that a

spectator pays when watching a game on a given date. Spectator may be students, adults, or seniors, with each category having its own charge rate. Give the cube definition using DMQL

6. Give the star schema for the base cuboid (student, test, attendance)

Concept Map



Syllabus

Introduction to Data warehouse – Concepts, Architecture, Data Models, Multidimensional Models - Data Cube, Concept Hierarchy & Schema Structures, OLAP operations, **Preprocessing -** Preprocessing Concepts, Data Cleaning, Integration, Transformation and Data Reduction, Data Discretization and Concept Hierarchy Generation, **Introduction to Data Mining –** Concepts, Functionality, DBMS vs Data mining, Issues, Challenges and Applications, **Association Techniques -** Introduction to Association Rules, Apriori algorithm, FP Growth , Mining on vertical data format, Correlation Analysis, Evaluation Metrics, **Classification Techniques-** Introduction to Classification **-** Constructing decision tree, Naive Bayes Classification, SVM Classification, Evaluation Metrics, Ensemble methods , Regression analysis , **Clustering Techniques -** Introduction to Clustering, Partitioning Method, Hierarchical Method, Density Based Method, Conceptual clustering, Outlier Analysis, **Applications of Data Mining -** Case studies on Temporal and Spatial Data Mining, Case studies on Web and Text Mining.

References

1. Jiawei Han, Micheline Kamper, Data Mining: Concepts and Techniques Morgan Kaufman, 2011, 3rd Edition, ISBN: 1-55860-489-8.
2. K.P.Soman, Shyam Diwakar, V.Ajay, "Insight into Data Mining – Theory and Practice", Prentice Hall of India, 2006.
3. Arun K.Pujari, "Data Mining Techniques", Universities Press, 2001.
4. 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 warehouse** (11)	
1.1	Concepts & Architecture	1
1.2	Data Models	2
1.3	Multidimensional Models	2
1.3.1	Data Cube & Concept Hierarchy	2
1.3.2	Schema Structures	2
1.4	OLAP operations	2
2.	**Preprocessing (6)**	
2.1	Preprocessing Concepts	2
2.2	Data Cleaning	2
2.3	Integration , Transformation and Data Reduction	1
2.4	Data Discretization and Concept Hierarchy Generation	1
3.	**Introduction to Data Mining (4)**	

No.	Topic	No. of Lectures
3.1	Concepts	1
3.2	Functionality	1
3.3	DBMS vs Data mining	1
3.4	Issues , Challenges and Applications	1
4	**Association Techniques(7)**	
4.1	Association Rules	1
4.2	Apriori algorithm	1
4.3	FP Growth tree	2
4.4	Mining on vertical data format	1
4.5	Correlation Analysis	1
4.6	Evaluation Metrics	1
5	**Classification Techniques (8)**	
5.1	Classification Concepts	1
5.2	Constructing decision tree – ID3 algorithm	2
5.3	Naive Bayes Classification	1
5.4	SVM Classification	1
5.5	Evaluation Metrics	1
5.6	Ensemble methods	1
5.7	Regression analysis	1
6	**Clustering Techniques (10)**	

No.	Topic	No. of Lectures
6.1	Clustering Concepts	1
6.2	Partitioning Method	1
6.3	Hierarchical Method	2
6.4	Density Based Method	2
6.5	Conceptual clustering	2
6.6	Outlier Analysis	2
7	**Applications of Data Mining (4)**	
7.1	Case studies on Temporal and Spatial Data Mining	2
7.2	Case studies on Web and Text Mining	2
	Total Periods	50

COURSE DESIGNER:

1. C. Deisy cdcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTH	4	0	-	4

CTH Agent Based Intelligent Systems 4:0

Preamble: This course introduces representations, techniques, and architectures used to build applied systems and to account for intelligence from a computational point of view.

Competencies

At the end of the Course, the Student will be able to

1. Understand the behavior of agents such as intelligent agents, logical agents, planning, and learning agent
2. Explain the various techniques applied for real time applications
3. Apply agent based probabilistic reasoning using Bayesian Networks
4. Apply Reinforcement learning for real time problems
5. Apply EM Algorithm and Hidden Markov Model

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. Define rational agent?
2. Define inference
3. What is Alpha Beta Pruning
4. Why Hidden Markov Model is used
5. Differentiate active and passive RL
6. What is mean by reward in RL
7. What is Bayesian learning
8. How EM is used for real time application

Understand

1. Differentiate single agent and multi agent
2. Apply Baye's Rule for real world problem
3. Explain First Order Inference

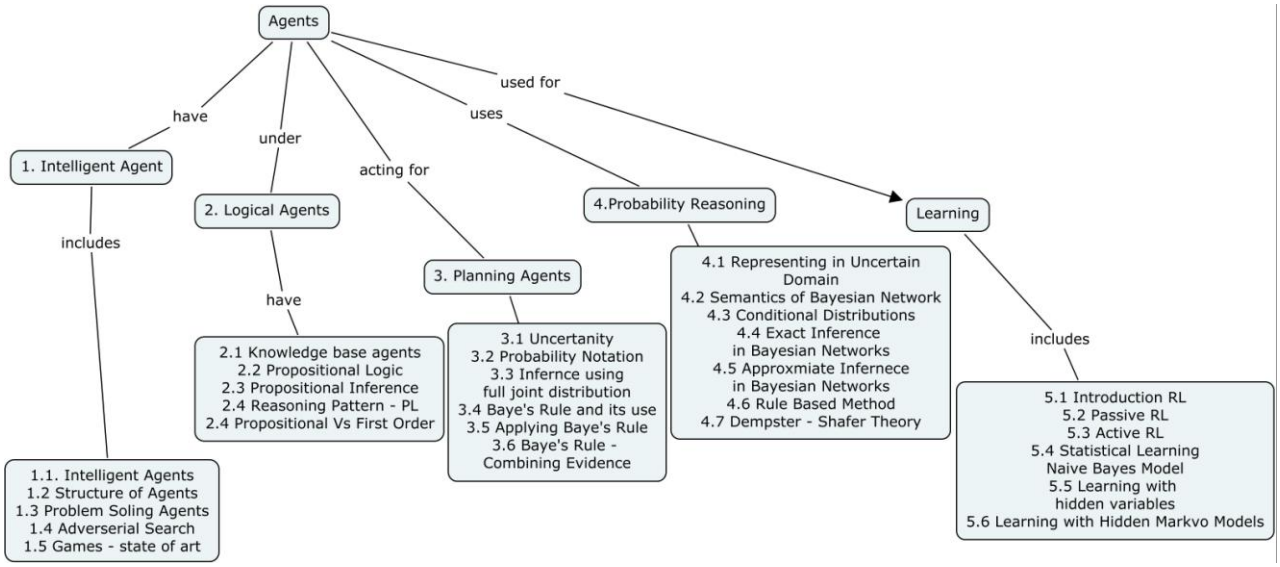
4. Explain Dempster-Shafer Theory
5. Explain Q Learning

7. Explain statistical learning with Naïve Baye's Models

Apply

1. Apply Optimal Strategies used for Game Problems
2. Demonstrate Agent Based Propositional Logic with diagram
3. Apply Baye's rule for combining evidence
4. Demonstrate the exact inference in Bayesian Networks
5. Demonstrate the learning in Bayesian networks with EM algorithm

Concept Map



Syllabus

Intelligent Agents : Introduction – Intelligent Agent – Structure of Agents - Problem Solving Agents- Searching for solutions – Adversarial Search – Games, Optimal strategies, Alpha-Beta Pruning, Imperfect Real Time decisions, Games that include an Element of Chance, State of the Art Games Programs **Logical Agents:** Knowledge Based Agents – Propositional Logic – Reasoning patterns in propositional logic – Effective propositional inference – Agent based Propositional Logic, Propositional versus First Order Inference

Planning Agents: Uncertainty – Acting under Uncertainty, Basic Probability Notation – Propositions, atomic events, prior probability, Axioms of Probability, Inference using Full Joint Distribution, Independence, Baye's Rule and its Use, Applying Baye's Rule, Using Baye's Rule: Combining Evidence **Probabilistic Reasoning:** Representing knowledge in an Uncertain Domain, Semantics of Bayesian Network, Efficient Representation of conditional distributions, Exact Inference in Bayesian Networks, Approximate Inference in

Bayesian Networks, Extending probability to first order representation, Rule based methods for uncertain reasoning, Dempster-Shafer Theory **Learning:** Introduction, Passive Reinforcement Learning, Active Reinforcement Learning, Statistical Learning – Learning with complete data, Naïve Baye's Models, Learning with Hidden Variables : EM Algorithm, Learning Hidden Markov Models

References

1. Stuart Russell," Artificial Intelligence – A Modern Approach" Third Edition, Pearson Edition,2003
2. Elaine Rich, Kevin knight, Shivashankar B Nair, " Artificial Intelligence" Third Edition, Tata Mc Graw Hill,2011

Course Contents and Lecture Schedule

No	Topic	No of Lectures
1.	**Intelligent Agent (10)**	
1.1	Introduction	2
1.2	Intelligent Agent, Structure of Agents	2
1.3	Problem Solving Agent, Searching for solutions	1
1.4	Adversarial Search – Games, Optimal strategies	1
1.5	Alpha-Beta Pruning	1
1.6	Imperfect Real Time decisions	1
1.7	Games that include an Element of Chance	1
1.8	State of the Art Games Programs	1
2.	**Logical Agents (10)**	
2.1	Introduction	1
2.2	Knowledge Based Agents	2
2.3	Propositional Logic	2
2.4	Reasoning Patterns in Propositional logic, Effective propositional inference	2
2.5	Agent Based Propositional Logic	1
2.6	Propositional versus First Order Inference	2
3.	**Planning Agents (10)**	
3.1	Uncertainty – Acting under Uncertainty	2
3.2	Basic Probability Notation – Propositions, atomic events, prior probability	2
3.3	Axioms of Probability	1

3.4	Inference using Full Joint Distribution	1
3.5	Independence	1
3.6	Baye's Rule and its Use	1
3.7	Applying Baye's Rule	1
3.8	Using Baye's Rule: Combining Evidence	1
4	**Probabilistic Reasoning**	
4.1	Representing knowledge in an Uncertain Domain	1
4.2	Semantics of Bayesian Network	2
4.3	Efficient Representation of conditional distributions	1
4.4	Exact Inference in Bayesian Networks	2
4.5	Approximate Inference in Bayesian Networks	1
4.6	Extending probability to first order representation	1
4.7	Rule based methods for uncertain reasoning	1
4.8	Dempster-Shafer Theory	1
5	**Learning (10)**	
5.1	Introduction	2
5.2	Passive Reinforcement Learning	1
5.3	Active Reinforcement Learning	1
5.4	Statistical Learning – Learning with complete data, Naïve Bayes Models	2
5.5	Learning with Hidden Variables : EM Algorithm	2
5.6	Learning Hidden Markov Models	2
	Total Periods	50

Course Designer

1. D. Tamilselvi dtamilselvi@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTJ	3	1	--	4

CTJ Cryptography and Network Security 3:1

Preamble: Cryptography is the science of information and communication security. It is used for authentication, encryption and access control. The objective of the course is to introduce some of the known security problems related to networking and to overview the contemporary solutions to these problems. The focus of the course is on Authentication, authorization, confidentiality, data integrity and non-repudiation. Network security and system security issues are also addressed here.

Prerequisite: CT14

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

1. Understand 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 corporate such as stock markets, banking etc.
5. Determine the strength of a given security algorithm when used in real time.

Assessment Pattern

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

Course Level Learning Objectives

Remember:

1. Distinguish between an attack and a threat.
2. Name the type of attack that is performed on a monoalphabetic substitution mechanism.

3. State the maximum number of rounds of operations that could be performed in AES?
4. State reasons for naming the Blowfish algorithm named so.
5. Define Golden Ratio.
6. List the uses of Firewalls?.

Understand:

1. Find the multiplicative inverse of (x^7+x+1) in mod $(x^8+x^4+x^3+x+1)$
2. Double DES is not used in practice. State reason
3. Perform the 2345>>>1ABC
4. Can the following matrix be used as a key in Hill cipher? Justify your answer.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

5. Compare and contrast a Hacker from a Cracker.
6. Generate the 8^{th} and 9^{th} round constants of AES.

Apply

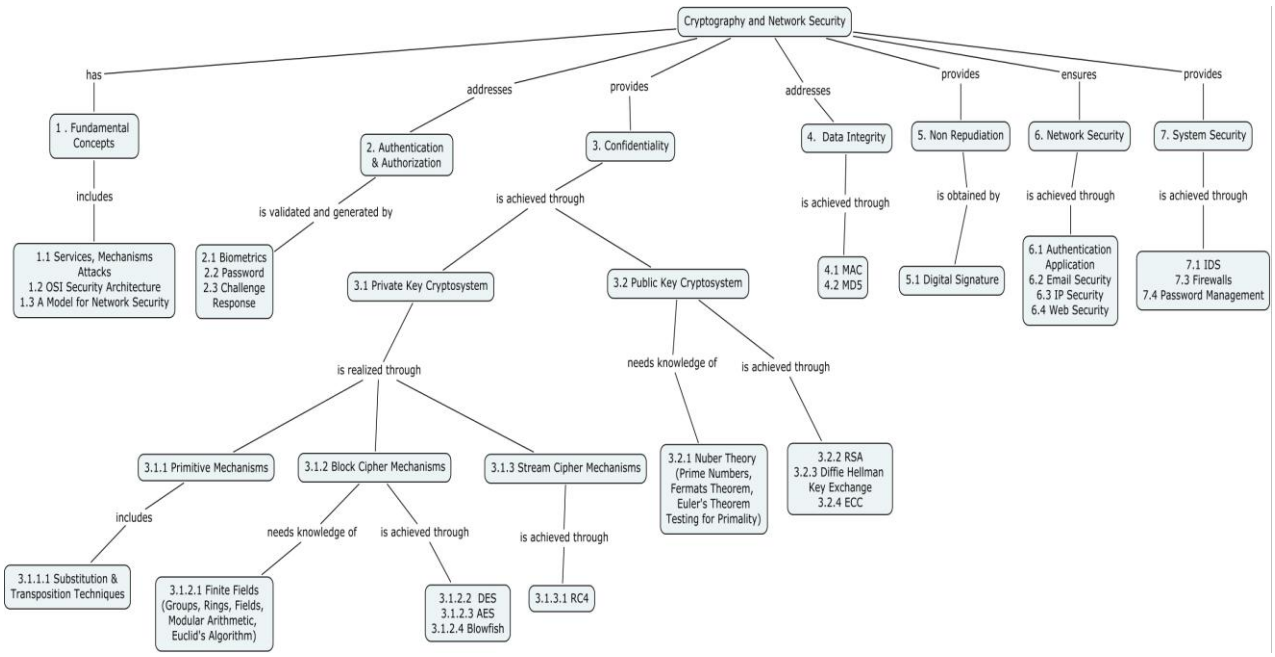
1. Apply a block cipher encryption mechanism where all the bits of the plaintext are treated equally and processed. The plain text is CRYPTOGRAPHICALGORITHM and key is SIXTY.
2. Perform two rounds of encryption for the plain text HOWAREYOU with the key FINE using a symmetric key encryption mechanism whose S box values are key dependant. Moreover this dependency makes it invulnerable to Brute Force attack.
3. Choose an appropriate encryption mechanism that can encrypt the message MULLAI PERIYAR DAM IS VERY STRONG with a key complexity of 26x26. Key of your own choice can be used. Shoe the decrypted information also.
4. Build an encryption mechanism with an auto key system to encrypt the plain text I HAVE DEBITED FIFTY CRORES FROM YOUR ACCOUNT. Use a key of your own choice. Work on the decryption mechanism also to get back the plain text.
5. Apply RSA algorithm to encrypt the message PAYRANSOM. Given p=19, q=23, e=7. Decrypt the message at the receiving end.
6. Apply PGP authentication and confidentiality services to give message M = 73A56F49257…. K_S=47524635. Given KU_b=5, KR_a=317 n_b=437, KU_a=11, KR_a=35 and n_a=221.

Analyze

1. Perform one round of cryptographic operations of DES on the plain text DATAMINING TECHNIQUES with key SOFT. Try to identify the relationship that exists between the encryption and decryption mechanisms.

2. Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key and send the result over the channel. Your partner XORs the incoming block with his key and sends it back. You check and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme? If so, what can be done to overcome this flaw?
3. Eve captures Bob's Hill cipher machine, which uses a 2-by-2 matrix M mod 26. She tries a chosen plaintext attack. She finds that the plaintext ba encrypts to HC and the plaintext zz encrypts to GT. What is the matrix M?
4. Caesar wants to arrange a secret meeting with Marc Antony, either at the Tiber (the river) or at the Coliseum (the arena). He sends the cipher text EVIRE. However, Antony does not know the key, so he tries all possibilities. Where will he meet Caesar?
5. A security system would like to provide message integrity using Message authentication code or MD5 mechanism. Which of the above mechanisms would you suggest? State reasons.
6. A video conferencing application would like to send its data more securely. Would you suggest a block cipher mechanism or a stream cipher mechanism for the same? State reasons.

Concept Map



Syllabus

Introduction: Services, Mechanisms and Attacks, The OSI Security Architecture, A Model for Network Security. **Authentication and Authorization:** Biometrics, Password, Challenge Response. **Confidentiality: Symmetric Ciphers:** Symmetric Cipher Model,

Substitution and Transposition Techniques. **Block Cipher Mechanisms:** DES, Block cipher modes of operation. Introduction to Finite Fields: Groups, Rings and Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields, Advanced Encryption Standard, Blowfish. **Stream Cipher Mechanism:** RC4 Stream Cipher. **Public Key Encryption**: Introduction to Number Theory: Prime Numbers, Fermat's and Euler's Theorem, Testing for Primality, RSA, Diffie – Hellman Key Exchange, Elliptic Curve Cryptography. **Message Authentication and Integrity:** Message Authentication Codes, MD5 Message Digest Algorithm. **Non-Repudiation:** Digital Signature and Digital Signature Standard. **Network Security Practice: Authentication Application**- Kerberos**. Electronic Mail Security -** PGP. **IP Security -** IP Security Architecture**. Web Security**- Secure Socket Layer and Transport layer, Secure Electronic Transaction. **System Security:** Intruders- Intrusion Detection, Password Management. Malicious Software - Viruses and related Threats. Firewalls- Firewal Design Principles.

References

1. William Stallings, Cryptography and Network Security Principles and Practices, Fourth Edition, Pearson Education, 2008.
2. Behrouz A. Foruzan, Cryptography and Network Security, TataMcGraw Hill, 2007
3. William Stallings, Cryptography and Network Security Principles and Practices, Third Edition, Pearson Education, 2006.

Course Contents and Lectures schedule

No	Topic	No of Lectures
1	**Fundamental Concepts (3)**	
1.1	Services, Mechanisms and Attacks	1
1.2	The OSI Security Architecture	1
1.3	A Model for Network Security.	1
2	**Authentication and Authorization (2)**	
2.1	Biometrics	1
2.2	Passwords	
2.3	Challenge Response Schemes	1
3	**Confidentiality (26)**	
3.1	Private Key Cryptosystem:	
3.1.1	**Primitive Mechanisms :** Symmetric Cipher Model	1
3.1.1.1	Substitution and Transposition Techniques.	3
3.1.2	**Block Cipher Mechanisms:**	

3.1.2.1	**Introduction to Finite Fields:** Groups, Rings Fields, Modular Arithmetic Euclid's Algorithm, Finite Fields	3
3.1.2.2	DES	2
3.1.2.3	Advanced Encryption Standard,	2
3.1.2.4	Blowfish	2
3.1.3	**Stream Cipher Mechanism:**	
3.1.3.1	RC4 Stream Cipher.	2
3.2	**Public Key Encryption**:	
3.2.1	Introduction to Number Theory: Prime Numbers, Fermat's and Euler's Theorem Testing for Primality	5
3.2.2	RSA	2
3.2.3	Diffie – Hellman Key Exchange	1
3.2.4	Elliptic Curve Cryptography	2
4	**Message Authentication and Integrity (4)**	
4.1	Message Authentication Codes	2
4.2	MD5 Message Digest Algorithm.	2
5	**Non-Repudiation (1)**	
5.1	Digital Signature and Digital Signature Standard.	1
6	**Network Security Practice** (8)	
6.1	**Authentication Application**- Kerberos**.**	2
6.2	**Electronic Mail Security-** PGP**.**	2
6.3	**IP Security-** IP Security Architecture**.**	2
6.4	**Web Security**- Secure Socket Layer and Transport layer, Secure Electronic Transaction	3
7	**System Security (6)**	
7.1	Intruders- Intrusion Detection,	1
7.2	Malicious Software- Viruses and related Threats.	2
7.3	Firewalls- Firewall Design Principles.	2
7.4	Password Management.	1
	Total	**50**

Course Designer:

P. Subathra pscse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTK	4	0	--	4

CTK – Design and Analysis of Parallel Algorithms 4:0

Preamble: This course aims at facilitating students to design and analyze parallel algorithms for fundamental problems in computer science. This course also provides the student with an understanding of parallelization frameworks like MPI, Open-MP using which these algorithms can be implemented.

Competencies

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

1. Design a parallel algorithm to solve a given problem.
2. Understand the design issues/trade-offs involved in designing a parallel algorithm.
3. Understand the different models of parallel machines.
4. Determine the asymptotic complexity of a parallel algorithm.
5. Apply basic parallelization techniques to design parallel algorithms for performing matrix operations, solving linear equations.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	10	10
2	Understand	30	30	30
3	Apply	50	40	40
4	Analyze	0	20	20
5	Evaluate	0	0	0
6	Create	0	0	0

Remember

1. State the three classes of parallel computation models.
2. State the difference between EREW and CREW SM SIMD computers.
3. Define a perfect shuffle interconnection pattern.
4. Define the cost of a parallel algorithm.
5. State the desirable properties of a parallel algorithm with respect to the no. of processors.
6. Describe the need for odd-even sorting networks.
7. State the purpose of the 'folding' stage while searching on a mesh.
8. State the worst case time complexity and cost of performing matrix transpose using a shuffle-connected computer.
9. State the reason behind the difficulty in analyzing MIMD algorithms.

10. State the idea behind sorting by conflict free merging.

Understand

1. Illustrate the different ways in which processors can be interconnected in a SIMD machine.
2. Describe a parallel algorithm to perform EREW sort and discuss the time complexity of this algorithm.
3. Paraphrase a parallel algorithm to perform mesh transpose.
4. Compare and contrast SIMD, SISD, MISD and MIMD computing models.
5. Illustrate how to simulate multiple accesses on a EREW computing model.
6. Summarize an algorithm to broadcast a datum to all the processors while executing an algorithm.
7. A tree connected computer with n leaves stores one integer of a sequence S per leaf. For a given k, 1<= k <= n, describe an algorithm that runs on this computer and selects the k^{th} smallest element of S.

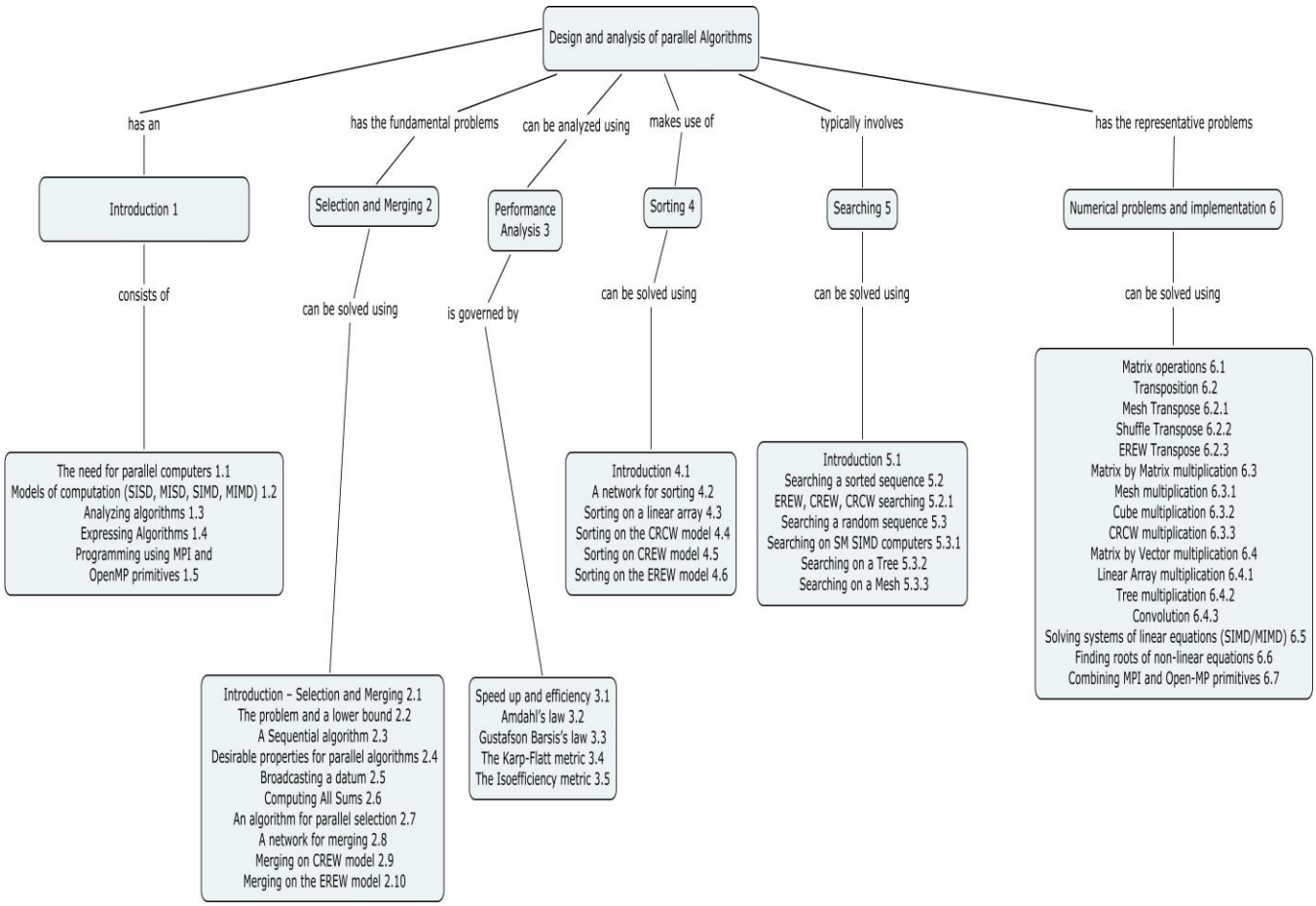
Apply

1. Given a set of numbers {s1, s2, … ,sn}, all sums of the form s1+s2, s1+s2+s3, s1+s2+…+sn are to be computed. Construct an algorithm for solving this problem using n processors on each of the four sub models of SM SIMD model.
2. Demonstrate that a fully connected network of n processors is equivalent to an EREW SM SIMD computer with n processors and exactly n locations of shared memory.
3. A satellite picture is represented as an n x n array of pixels each taking an integer value between 0 and 9, thus providing various gray levels. It is required to smooth the picture, that is the value of pixel (i,j) is to be replaced by the average of its value and those of its eight neighbors. Illustrate a special purpose parallel architecture to solve this problem. Assume that m the number of processors available is less than n^2 the no. of pixels. Construct two different implementations of the smoothing process and analyze their running times.
4. Construct a parallel merging algorithm for the CRCW model.
5. Illustrate the MIMD sorting by enumeration algorithm for the following set of inputs: S={8,6,6,9,7} and no. of processors=2
6. A q-dimensional cube connected SIMD computer with n = 2^q processors P0, P1, …, Pn-1 is given. Each processor Pi holds a datum xi. Construct a parallel algorithm to replace x0 with x0+x1+ … + xn-1
7. Construct a parallel algorithm to find roots of non-linear equations using Newton-Raphson method.

Analyze

1. Analyze the worst case time complexity of the parallel selection algorithm
2. Analyze the worst case time complexity to perform matrix transpose on a mesh
3. Analyze the suitability of each of the SM SIMD models to solve the systems of linear equations using a parallel algorithm
4. Analyze the worst case time complexity to find the roots of non-linear equations using a parallel algorithm based on Newton-Raphson's method
5. Analyze the implications and advantages of combining MPI and OpenMP to implement parallel algorithms
6. Analyze and compare the worst case time complexities of different algorithms to perform searching on a random sequence using different models of SM SIMD machines
7. Analyze the worst case time complexity to broadcast a datum on an EREW SM SIMD computer.

Concept Map



Syllabus

Introduction: The need for parallel computers, Models of computation (SISD, MISD, SIMD, MIMD), Analyzing algorithms, Expressing Algorithms, Programming using MPI and Open-MP primitives. **Selection and Merging:** Introduction – Selection and Merging, The problem and a lower bound, A Sequential algorithm, Desirable properties for parallel algorithms, Broadcasting a datum, Computing All Sums, An algorithm for parallel selection, A network for merging, Merging on CREW model, Merging on the EREW model. **Performance Analysis:** Speed up and efficiency, Amdahl's law, Gustafson Barsis's law, The Karp-Flatt metric, The Isoefficiency metric. **Sorting:** Introduction, A network for sorting, Sorting on a linear array, Sorting on the CRCW model, Sorting on CREW model, Sorting on the EREW model. **Searching:** Introduction, Searching a sorted sequence, EREW, CREW, CRCW searching, Searching a random sequence, Searching on SM SIMD computers, Searching on a Tree, Searching on a Mesh. **Numerical problems and implementation**: Matrix operations, Transposition, Mesh Transpose, Shuffle Transpose, EREW Transpose, Matrix by Matrix multiplication, Mesh multiplication, Cube multiplication, CRCW multiplication, Matrix by Vector multiplication, Linear Array multiplication, Tree multiplication, Convolution, Solving systems of linear equations (SIMD/MIMD), Finding roots of non-linear equations, Combining MPI and Open-MP primitives.

References

1. S.G. Akl, "The design and analysis of parallel algorithms", Prentice Hall of India, 1989.
2. Michael Jay Quinn, "Parallel programming in C with MPI and OpenMP", McGraw-Hill Higher Education, 2004.
3. S. Lakshmivarahan and S.K. Dhall, "Analysis and design of parallel algorithms – Arithmetic and Matrix problems", McGraw Hill, 1990.

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Introduction**	
1.1	The need for parallel computers	1
1.2	Models of computation (SISD, MISD, SIMD, MIMD)	1
1.3	Analyzing algorithms	1
1.4	Expressing Algorithms	1
1.5	Programming using MPI and Open-MP primitives	2
2	**Selection and Merging**	

No.	Topic	No. of Lectures
2.1	Introduction – Selection and Merging	1
2.2	The problem and a lower bound	1
2.3	A Sequential algorithm	1
2.4	Desirable properties for parallel algorithms	1
2.5	Broadcasting a datum	1
2.6	Computing All Sums	1
2.7	An algorithm for parallel selection	1
2.8	A network for merging	1
2.9	Merging on CREW model	1
2.10	Merging on the EREW model	1
3	**Performance Analysis**	
3.1	Speed up and efficiency	1
3.2	Amdahl's law	1
3.3	Gustafson Barsis's law	1
3.4	The Karp-Flatt metric	1
3.5	The Isoefficiency metric	1
4	**Sorting**	
4.1	Introduction	1
4.2	A network for sorting	1
4.3	Sorting on a linear array	1
4.4	Sorting on the CRCW model	1
4.5	Sorting on CREW model	1
4.6	Sorting on the EREW model	1
5	**Searching**	
5.1	Introduction	1
5.2	Searching a sorted sequence	1

No.	Topic	No. of Lectures
5.2.1	EREW, CREW, CRCW searching	1
5.3	Searching a random sequence	1
5.3.1	Searching on SM SIMD computers	1
5.3.2	Searching on a Tree	1
5.3.3	Searching on a Mesh	1
6	**Numerical problems and implementation**	
6.1	Matrix operations	1
6.2	Transposition	1
6.2.1	Mesh Transpose	1
6.2.2	Shuffle Transpose	1
6.2.3	EREW Transpose	1
6.3	Matrix by Matrix multiplication	1
6.3.1	Mesh multiplication	1
6.3.2	Cube multiplication	1
6.3.3	CRCW multiplication	0.5
6.4	Matrix by Vector multiplication	1
6.4.1	Linear Array multiplication	1
6.4.2	Tree multiplication	1
6.4.3	Convolution	1
6.5	Solving systems of linear equations (SIMD/MIMD)	1
6.6	Finding roots of non-linear equations	1
6.7	Combining MPI and Open-MP primitives	1.5

COURSE DESIGNERS

1. S. Karthick, skcse@tce.edu
2. S. Mercy Shalinie shalinie@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTL	4	0	--	4

CTL – Distributed and Grid Computing 4:0

Preamble: This course aims at facilitating students to understand the design issues and implementation details behind distributed computing and grid computing middlewares. The prerequisites for this course are operating systems and networks. A previous course on Distributed systems would be an added advantage.

Competencies

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

1. Understand the hardware and software building blocks of a distributed system.
2. Understand the design issues involved in designing a distributed/grid system in terms of networking/OS/middleware.
3. Understand the limitations of a distributed system like absence of a global state etc.
4. Understand and apply the various distributed transactions and distributed mutual exclusion schemes available to design an effective distributed system.
5. Apply the SOA principles to design a computational/data grid.
6. Understand the design trade-offs involved in selecting a grid computing toolkit and to identify a toolkit to be used for a specific application.
7. Understand the importance of agreement protocols and election algorithms in designing a distributed system.

Assessment 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

Remember

1. Define point-to-point security in the context of grid services.
2. Define Service Level Agreement (SLA)
3. State the purpose behind autonomic computing
4. Define a computational grid
5. Define location transparency
6. State the difference between a stub and a dynamic proxy.

7. State the reasoning behind service virtualization.
8. Define utility computing.
9. State the basic principles behind autonomic computing
10. Define a semantic grid

Understand

1. Illustrate the benefits of using ontologies for resource discovery in semantic grids.
2. Compare and contrast grid computing with cluster computing and P2P systems.
3. Illustrate how Hadoop achieves data locality optimization while executing a MR algorithm.
4. Compare and contrast grid services and web services.
5. Illustrate the architecture of openstack cloud framework with a neat sketch.
6. Illustrate the functionalities provided by the layered grid architecture with a neat sketch.
7. Illustrate the benefits of using Nimrod for implementing parametric applications

Apply

1. Construct a policy expression and attachment to satisfy the below mentioned requirement:

   A trading grid service is running in NYSE (New York Stock Exchange) which accepts trade requests for selling/buying a stock. Before a trade can be carried out the endpoint wants to make sure that the user is authenticated using a Kerberos V5 ticket or a X.509 certificate. The portion of the message carrying the stock symbol and quantity to be sold/bought should be kept confidential. The end-point would also like to ensure that the above mentioned XML fragment is not altered while the message is in transit.

2. a. Illustrate the trust establishment process with respect to the following grid application:

   A user from the domain tce.edu wants to collaborate and work with a user from the domain stanford.edu. Both of these domain controllers trust the Verisign certificate authority.

   b. Identify the discrepancies/issues which might arise between the domains after the trust establishment process is complete and before the communication begins.

   c. Illustrate the role of the higher level standards in the WS-Security stack, in resolving the above mentioned inter-domain discrepancies.

3. Design a data grid to solve the following problem using a *Map-Reduce* algorithm:

   A huge text document (~100 PB) needs to be analyzed for the frequency of occurrence of words present within the document. This analysis has to be done after removing the stop-words like {articles (a, an, the), prepositions}. The

expected output is the list of words along with their frequency of occurrence. Compose snippets of code for mapper and reducer functions, using one of the languages mentioned below: {Java/python/Ruby/C++}.

4. 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.
5. a. Construct a SOAP module to satisfy the below mentioned security requirements:

   A banking firm has exposed a grid service, which users can contact to download and view account statements (contains credits/debits which took place in the account within a specific time period). Before downloading the account statement the end points have to mutually authenticate each other through X.509 digital certificates. The integrity of the messages transferred should be checked at both the client and the server.

   b. Illustrate the need for canonicalization algorithms while checking the message integrity.
6. Construct the task-machine mapping by executing the sufferage heuristic for the task set, T = {t1, t2, t3}. Assume that there are four machines in the grid M = {m1, m2, m3, m4}. The predicted execution time (in sec) of the tasks in T on machines in M is given by the following table:

Tasks/machines	m1	m2	m3	m4
t1	3	5	11	10
t2	3	2	3	4
t3	5	8	16	21

   The machine availability time (MAT) of the machines in M is given by the following table:

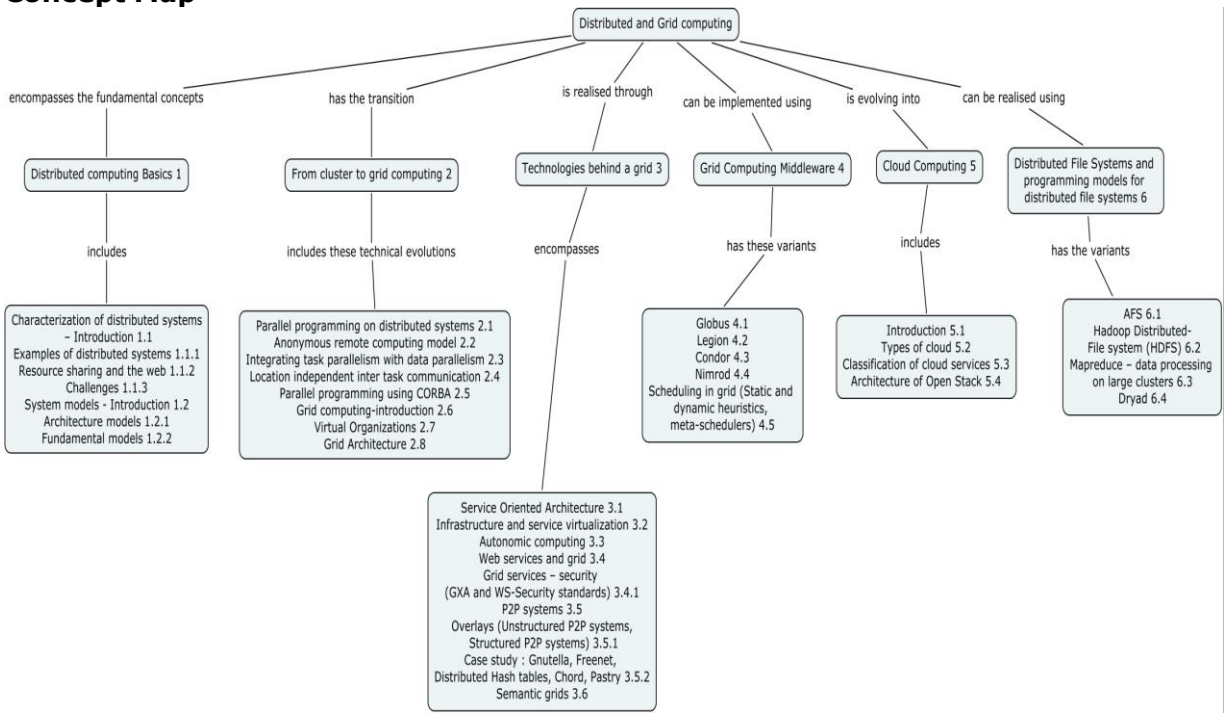
machine	m1	m2	m3	m4
MAT (in sec)	8	2	3	5

7. Consider the following Virtual Organization (VO) formed for weather prediction:

This VO requires resources such as weather prediction software applications, to perform mandatory environment simulations associated with predicting weather. Likewise they will require very specific hardware resources to run the respective software as well as high speed data storage facilities to maintain the data generated from performing simulations.

Identify the architectural requirements to be satisfied by the following grid components: a. Resource categories b. VO c. User applications

Concept Map



Syllabus

Introduction – Distributed computing: Characterization of distributed systems – Introduction, Examples of distributed systems, Resource sharing and the web, Challenges, System models – Introduction, Architecture models, Fundamental models. **From cluster to grid computing:** Parallel programming on distributed systems, Anonymous remote computing model, Integrating task parallelism with data parallelism, Location independent inter task communication, Parallel programming using CORBA, Grid computing-introduction, Virtual Organizations, Grid Architecture. **Technologies behind a grid**: Service Oriented Architecture, Infrastructure and service virtualization, Autonomic computing, Web services and grid, Grid services – security (GXA and WS-Security standards), P2P systems – Overlays (Unstructured P2P systems, Structured P2P systems), Case study : (Gnutella, Freenet, Distributed Hash tables, Chord, Pastry), Semantic grids. **Grid Computing Middleware**: Globus, Legion, Condor, Nimrod, Scheduling in grid (Static and Dynamic heuristics, meta-schedulers). **Cloud Computing:** Introduction, Types of cloud, Classification of cloud services, Architecture of Open Stack. **Distributed File Systems and programming models for distributed file systems:** AFS, Hadoop Distributed File system (HDFS), Mapreduce – data processing on large clusters, Dryad.

References

1. George Coulouris, Jean Dollimore, Tim Kindberg: "Distributed Systems – concepts and design", Third edition, Pearson Education, 2001.
2. Joshy Joseph, Craig Fellenstein: "Grid Computing", First Edition, Pearson Education, 2004.
3. Ian Foster and Carl Kesselman: "The Grid2 - Blueprint for a new computing infrastructure", Second Edition, Morgan Kaufmann, 2006.
4. Andrew S. Tanenbaum: "Distributed Operating Systems", First Edition, Pearson Education, 2008.
5. Mukesh Singhal, Niranjan G. Shivaratri: "Advanced concepts in Operating Systems", First Edition, Tata McGraw-Hill, 2006.
6. Andrew S. Tanenbaum, Maarten Van Steen: "Distributed Systems – Principles and Paradigms", Second Edition, Pearson Education, 2008.
7. Maozhen Li, Mark Baker: "The Grid Core Technologies", First Edition, John Wiley & Sons, 2005.
8. D. Janakiram: "Grid Computing", Tata Mc-Graw Hill, First edition, 2005.
9. Tom White: "Hadoop: The definitive guide Map reduce for the cloud", O'Reilly media, First edition, 2009.
10. Frederic Magoules, Thi-Mai-Huong Nguyen, Lei Yu: "Grid Resource Management: Toward Virtual and Services Compliant Grid Computing", First Edition, CRC Press, 2008.
11. http://openstack.org/

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Distributed computing Basics**	
1.1	Characterization of distributed systems – Introduction	1
1.1.1	Examples of distributed systems	1
1.1.2	Resource sharing and the web	1
1.1.3	Challenges	1
1.2	System models – Introduction	1
1.2.1	Architecture models	1
1.2.2	Fundamental models	1

No.	Topic	No. of Lectures
2	**From cluster to grid computing**	
2.1	Parallel programming on distributed systems	2
2.2	Anonymous remote computing model	2
2.3	Integrating task parallelism with data parallelism	2
2.4	Location independent inter task communication	2
2.5	Parallel programming using CORBA	2
2.6	Grid computing-introduction	2
2.7	Virtual Organizations	1
2.8	Grid Architecture	2
3	**Technologies behind a grid**	
3.1	Service Oriented Architecture	2
3.2	Infrastructure and service virtualization	1
3.3	Autonomic computing	1
3.4	Web services and grid	1
3.4.1	Grid services – security (GXA and WS-Security standards)	2
3.5	P2P systems	1
3.5.1	Overlays (Unstructured P2P systems, Structured P2P systems)	1
3.5.2	Case study : Gnutella, Freenet, Distributed Hash tables, Chord, Pastry	2
3.6	Semantic grids	0.5
4	**Grid Computing Middleware**	
4.1	Globus	1
4.2	Legion	1
4.3	Condor	1
4.4	Nimrod	0.5

No.	Topic	No. of Lectures
4.5	Scheduling in grid (Static and dynamic heuristics, meta-scheduling schemes)	3
5	**Cloud Computing**	
5.1	Introduction	0.5
5.2	Types of cloud	0.5
5.3	Classification of cloud services	0.5
5.4	Architecture of Open Stack	0.5
6	**Distributed File Systems and programming models for distributed file systems**	
6.1	AFS	2
6.2	Hadoop Distributed File system (HDFS)	2
6.3	Mapreduce – data processing on large clusters	2
6.4	Dryad	2

COURSE DESIGNER:

1. S. Karthick skcse@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CT M	4		-	4

CTM Software Reliability 4:0

Preamble: software reliability provides general introduction to software reliability engineering. It presents detailed analytical models, state-of-the-art techniques, and methodologies used to assess the reliability of software systems.

Prerequisite: CT22

Competencies

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

1. Understand the basics of reliability engineering
2. Differentiate software reliability and hardware reliability
3. Apply various reliability models for given software.
4. Apply various cost models for given software.
5. Apply various Fault-Tolerant techniques for given software.

Assessment Pattern

	Bloom's Category	Test 1	Test 2	Test 3/End-semester examination
1	Remember	20	20	20
2	Understand	40	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 are the needs for software reliability
2. Differentiate software reliability and hardware reliability
3. Differentiate verification with validation.
4. State the differentiate between Maintainability and availability.
5. Define "cyclomatic complexity metric"
6. What is meant by Recovery block?
7. List the Environmental factors that affect the reliability

Understand

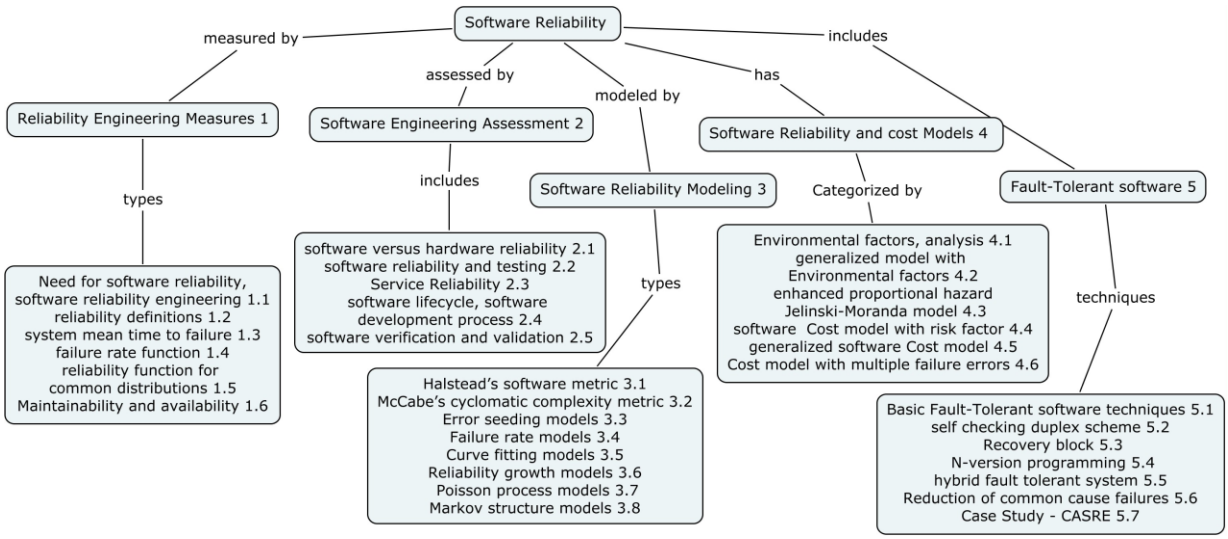
1. How does the saturation effect affect the reliability estimate of a program?
2. When using a "white-box" based approach to reliability estimation. when is a test case considered useful?

3. Compare Error seeding models with Failure rate models
4. Explain Basic Fault-Tolerant software techniques in detail
5. Explain how to generate generalized reliability model with Environmental factors
6. Discuss about various software cost models in detail

Apply

1. Assuming failures on different processors are independent, apply a Markov model based on the failure and recovery rates.(assume own data)
2. Determine the amount of time by which the mean value function during the second stage of integration is translated from the mean value function that would have been observed had the entire system been in place at the start of testing.
3. Why do combination models provide better results, on average, than individual models? Suggest list of other methods to increase the prediction accuracy of models?
4. Compare various failure rate models and suggest a model when failures on different processors are dependent
5. Apply N-version programming for hybrid fault tolerant system
6. Solve the Reliability growth model to obtain the reliability of the three-out-of-five system

Concept Map



Syllabus

Reliability Engineering Measures: Need for software reliability, software reliability engineering, reliability definitions, system mean time to failure, failure rate function, reliability function for common distributions, Maintainability and availability. **Software Engineering Assessment:** software versus hardware reliability, software reliability and testing, Service Reliability, software lifecycle, software development process, software

verification and validation. **Software Reliability Modeling:** Halstead's software metric, McCabe's cyclomatic complexity metric, Error seeding models, Failure rate models, Curve fitting models, Reliability growth models, Poisson process models, Markov structure models. **Software Reliability and cost Models:** Environmental factors, analysis, generalized model with Environmental factors, enhanced proportional hazard Jelinski-Moranda model, software Cost model with risk factor, generalized software Cost model, Cost model with multiple failure errors. **Fault-Tolerant software:** Basic Fault-Tolerant software techniques, self checking duplex scheme, Recovery block, N-version programming, hybrid fault tolerant system, Reduction of common cause failures, Case Study - CASRE

References

1. Hoang Pham, "system Software Reliability", First Edition , Springer-Verlag, 2006
2. John D. Musa, "Software Reliability Engineering", Tata McGraw Hill, 2004.
3. Doron A. Peled "Software Reliability Methods" Springer-Verlag, 2001

Course Contents and Lecture Schedule

No	Topic	No of Lectures
1.	**Reliability Engineering Measures (7)**	
1.1	Need for software reliability, software reliability engineering	1
1.2	Reliability definitions	1
1.3	System mean time to failure	1
1.4	Failure rate function	1
1.5	Reliability function for common distributions	2
1.6	Maintainability and availability	1
2.	**Software Engineering Assessment(6)**	
2.1	Software versus hardware reliability	1
2.2	Software reliability and testing	1
2.3	Service Reliability	2
2.4	Software lifecycle, software development process	1
2.5	Software verification and validation	1

3.	**Software Reliability Modeling(16)**	
3.1	Halstead's software metric	1
3.2	Mccabe's cyclomatic complexity metric	1
3.3	Error seeding models	2
3.4	Failure rate models	3
3.5	Curve fitting models	2
3.6	Reliability growth models	3
3.7	Poisson process models	2
3.8	Markov structure models	2
4	**Software Reliability and cost Models(11)**	
4.1	Environmental factors, analysis	2
4.2	Generalized model with Environmental factors	2
4.3	Enhanced proportional hazard Jelinski-Moranda model	2
4.4	Software Cost model with risk factor	2
4.5	Generalized software Cost model	1
4.6	Cost model with multiple failure errors	2
5	**Fault-Tolerant software(10)**	
5.1	Basic Fault-Tolerant software techniques	1
5.2	Self checking duplex scheme	1
5.3	Recovery block	1
5.4	N-version programming	2
5.5	Hybrid fault tolerant system	2
5.6	Reduction of common cause failures	1
5.7	Case Study - CASRE	2
	Total Periods	**50**

Course Designer

A. Malini amcse@tce.edu

Board of studies meeting on 07.04.2012 Approved in 44th Academic council meeting on 09.06.2012

Sub Code	Lectures	Tutorial	Practical	Credit
CTN	4	-	-	4

CTN Agile Testing 4:0

Preamble: Agile testing illustrate the tester's role with examples from real agile teams and explains how to use the agile testing quadrants to identify what testing is needed, who should do it, and what tools might help.

Competencies

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

1. Understand agile methods, principles and software development.
2. Understand and differentiate various agile process models.
3. Differentiate traditional and agile testing.
4. Apply agile principles and values to testing.
5. Apply mutation testing for a given program.
6. Understand the Quadrants of Agile Testing.
7. Automate the test process.

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	60	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 meant by agility?
2. List agile manifesto and principles.
3. What is meant by pair programming?
4. Differentiate traditional teams and Agile teams
5. What are the purposes of testing?
6. List the barriers to automation.

Understand

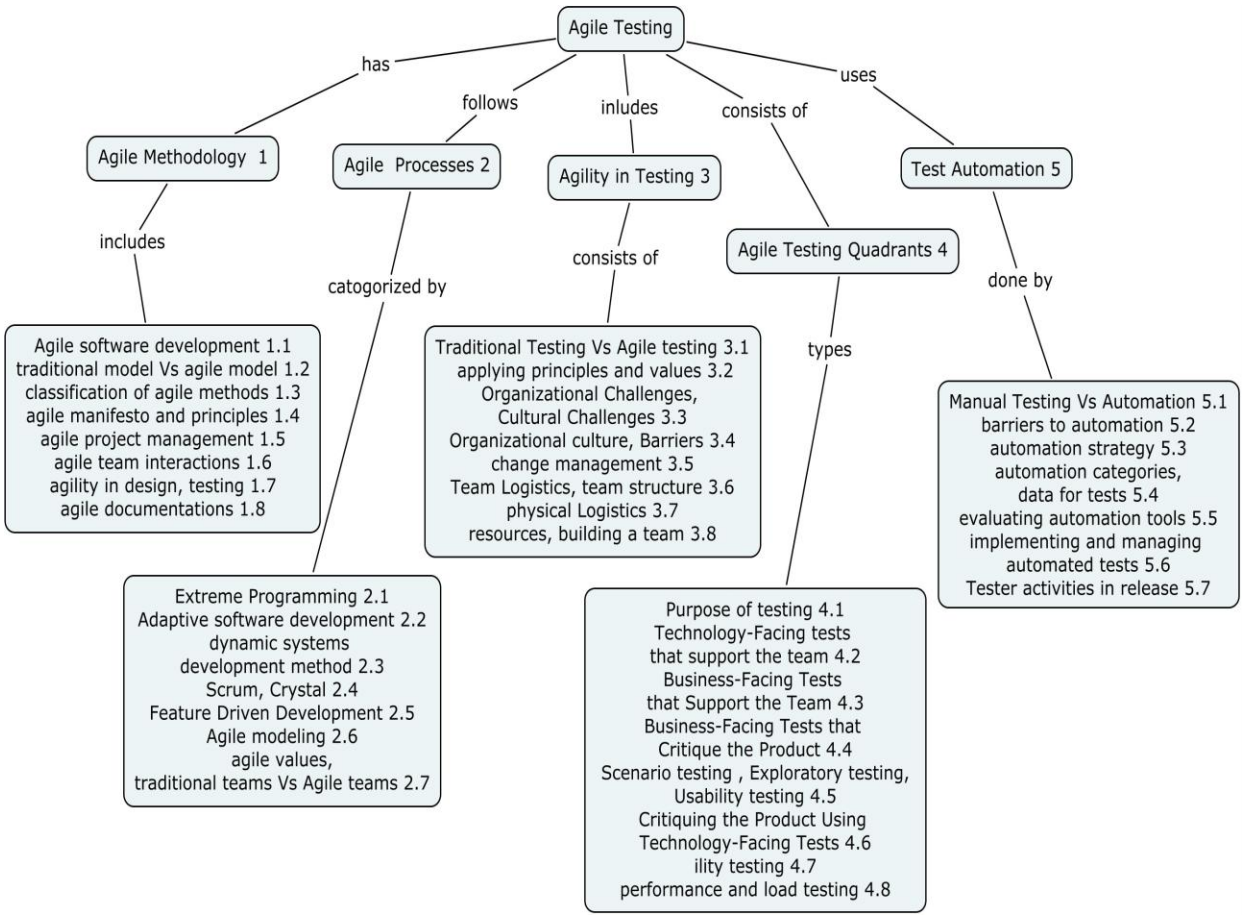
1. Explain the various agile methods in detail.
2. Explain about Feature Driven Development.

3. Discuss about Team Logistics.
4. Explain Scenario testing and Exploratory testing in detail
5. Explain performance and load testing.
6. Summarize the activities of tester during software release.

Apply

1. Classify various agile process models highlighting their advantages and disadvantages.
2. Contrast Adaptive software development and dynamic systems development method.
3. Distinguish Traditional Testing and agile testing.
4. Illustrate a successful adoption of agile testing over Organizational and Cultural Challenges.
5. Demonstrate the significance 'ility testing' with an example.
6. Identify the test automation strategy to automate the tests for medium sized software.

Concept Map



Syllabus

Agile Methodology: Agile software development, traditional model Vs agile model, classification of agile methods, agile manifesto and principles, agile project management,

agile team interactions, agility in design, testing, agile documentations. **Agile Processes:** Extreme Programming, Adaptive software development, dynamic systems development method, Scrum, Crystal, and Feature Driven Development, Agile modeling, agile values, traditional teams Vs Agile teams **Agility in Testing:** Traditional Testing Vs Agile testing, applying principles and values, Organizational Challenges, Cultural Challenges, Organizational culture, Barriers, change management, Team Logistics, team structure, physical Logistics, resources, building a team. **Agile Testing Quadrants:** Purpose of testing, Technology-Facing tests that support the team, Business-Facing Tests that Support the Team, Business-Facing Tests that Critique the Product, Scenario testing , Exploratory testing, Usability testing, Critiquing the Product Using Technology-Facing Tests, ility testing, performance and load testing. **Test Automation:** Manual Testing Vs Automation, barriers to automation, automation strategy, automation categories, data for tests, evaluating automation tools, implementing and managing automated tests, Tester activities in release.

References

1. Lisa Crispin, Janet Gregory, "Testing: A Practical Guide for Testers and Agile Teams ", First Edition ,Pearson Education, 2010
2. Craig Larman, "Agile and Iterative Development: A manager's Guide", First Edition, Pearson Education, 2007.
3. Roger S. Pressman, Software Engineering A Practitioner's Approach, Seventh Edition, Mcgraw Hill International Edition.,2010
4. William E. Perry, "Effective methods for software testing", Second Edition, John wiley & Sons,2000

Course Contents and Lecture Schedule

No	Topic	No of Lectures
1.	**Agile Methodology (8)**	
1.9	Agile software development	1
1.10	traditional model Vs agile model	1
1.11	classification of agile methods	1
1.12	agile manifesto and principles	1
1.13	agile project management	1
1.14	agile team interactions	1
1.15	agility in design, testing	1
1.16	agile documentations	1
2.	**Agile Processes (7)**	
2.7	Extreme Programming	1

2.8	Adaptive software development	1
2.9	dynamic systems development method	1
2.10	Scrum, Crystal,	1
2.11	Feature Driven Development	1
2.12	Agile modeling	1
2.13	agile values, traditional teams Vs Agile teams	1
3.	**Agility in Testing (10)**	
3.9	Traditional Testing Vs Agile testing	1
3.10	applying principles and values	2
3.11	Organizational Challenges, Cultural Challenges	1
3.12	Organizational culture, Barriers	2
3.13	change management	1
3.14	Team Logistics, team structure	1
3.15	physical Logistics	1
3.16	Resources, building a team.	1
4	**Agile Testing Quadrants (10)**	
4.9	Purpose of testing	1
4.10	Technology-Facing tests that support the team	1
4.11	Business-Facing Tests that Support the Team	1
4.12	Business-Facing Tests that Critique the Product	2
4.13	Scenario testing , Exploratory testing, Usability testing	1
4.14	Critiquing the Product Using Technology-Facing Tests	2
4.15	ility testing, performance and load testing	1
4.16	Resources, building a team	1
5	**Test Automation (10)**	
5.7	Manual Testing Vs Automation	1
5.8	barriers to automation	1
5.9	automation strategy	2
5.10	automation categories, data for tests	1
5.11	evaluating automation tools	1
5.12	implementing and managing automated tests	2
5.13	Tester activities in release	2
	Total	**45**

Course Designer

1. N. ShivaKumar shiva@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTP	4	0	-	4

CTP Cloud Computing 4:0

Preamble: This course is offered as an elective for the Post Graduate students of Computer Science and Engineering. This course is aimed at introducing cloud computing, the services offered by the cloud, building cloud networks, virtualization, distributed storage and security.

Competencies

At the end of the Course, the students will be able to

1. Explain the evolution of cloud computing and the web services offered.
2. Explain the building of a cloud network.
3. Discuss in detail about Virtualization.
4. Discuss the importance of security in the cloud.
5. Explain cloud storage.

Assessment Pattern

	Bloom's Category	**Test 1**	**Test 2**	**Test 3**	**End-semester examination**
1	Remember	30	20	20	20
2	Understand	30	30	20	20
3	Apply	40	50	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. What are the key characteristics of cloud computing?
2. What is a cloud data center?
3. List the benefits of virtualization.
4. What are the security challenges in cloud computing?
5. What is data footprint reduction?

Understand

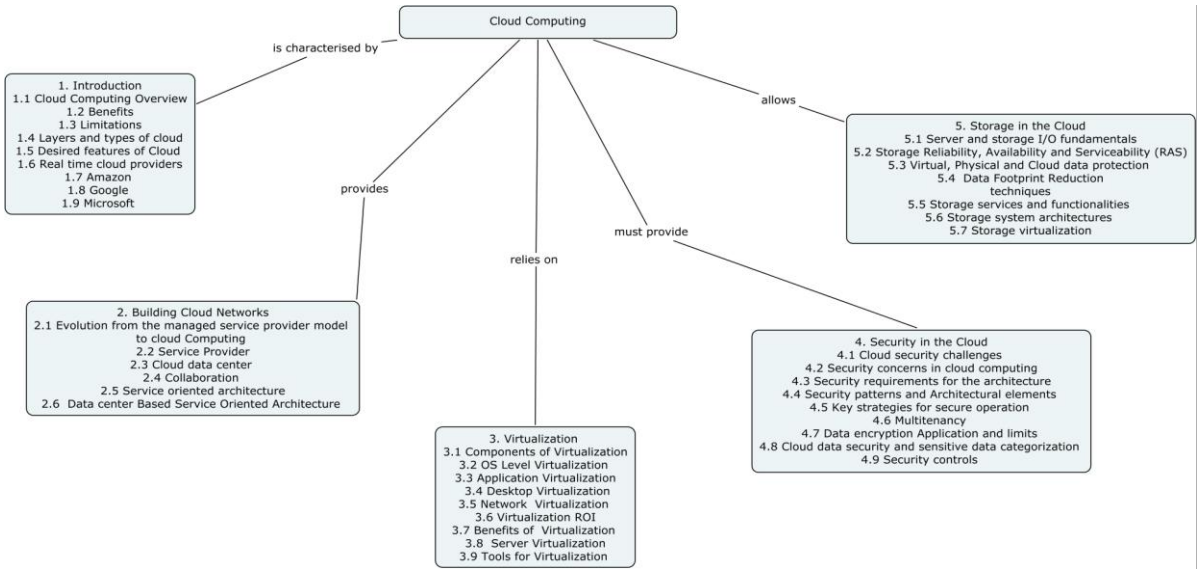
1. Describe in detail the web services delivered from the cloud.
2. Explain in detail about Data center Based Service Oriented Architecture.
3. Discuss in detail the different types of virtualization.

4. Explain sensitive data categorization to provide cloud data security.
5. Discuss in detail about the DFR techniques.

Apply

1. Apply Monitoring-As-A-Service to real time log monitoring.
2. Explain how your company can build highly automated private cloud networks that can be managed from a single point.
3. Assume that a company ABC wants to offer services such as starting a policy and payment of policy on a service oriented architecture over the cloud. Explain the implementation details of this scenario.
4. Consider a Ubuntu virtual machine installed over a Windows machine. Explain the SNAT and DNAT configuration that needs to be applied to enable the Ubuntu virtual machine communication with the external network and the external network communication with the Ubuntu virtual machine.
5. Explain how the DFR techniques can be applied to your cloud to provide capacity optimization.

Concept Map:



Syllabus

Introduction – Cloud Computing Overview – benefits – limitations – Layers and types of cloud – Desired features of Cloud – Real time cloud providers – Amazon – Google – Microsoft - **Building Cloud Networks -** Evolution from the managed service provider model to cloud Computing - Service Provider - Cloud data center – Collaboration - Service oriented architecture - Data center Based Service Oriented Architecture **Virtualization -** Components of Virtualization - OS Level Virtualization - Application Virtualization - Desktop Virtualization - Network Virtualization - Virtualization ROI-Benefits of Virtualization - Server Virtualization – Tools for Virtualization - **Security in the Cloud -**

Cloud security challenges - Security concerns in cloud computing - Security requirements for the architecture - Security patterns and Architectural elements - Key strategies for secure operation – Multitenancy – Data encryption Application and limits – Cloud data security and sensitive data categorization – Security controls – **Storage in the Cloud –** Server and storage I/O fundamentals – Storage Reliability, Availability and Serviceability (RAS) - Virtual, Physical and Cloud data protection - Data Footprint Reduction techniques – Storage services and functionalities – Storage system architectures – Storage virtualization

References:

1. Rajkumar Buyya, James Brogerg, Andrzej Goscinski, "Cloud Computing Principles and Paradigms", John Wiley & Sons, Inc. Publications, 2011.
2. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter,"Cloud Computing:A Practical Approach", McGraw Hill Publication, 2010.
3. John Rittinghouse, James F. Ransome, "Cloud Computing: Implementation, Management and Security", CRC Press 2010.
4. Vic (J.R.) Winkler, "Securing the Cloud: Cloud Computer Security Techniques and Tactics, Elsevier, 2011
5. Greg Schulz, "Cloud and Virtual Data Storage Networking", CRC Press, 2012.

Course contents and Lecture Schedule

No	Topic	No. of Lectures
1	**INTRODUCTION (12)**	
1.1	Cloud Computing Overview	1
1.2	Benefits	1
1.3	Limitations	1
1.4	Layers and types of cloud	1
1.5	Desired features of Cloud	1
1.6	Real time cloud providers	1
1.7	Amazon	2
1.8	Google	2
1.9	Microsoft	2
2	**BUILDING CLOUD NETWORKS (10)**	
2.1	Evolution from the managed service provider model to cloud Computing	2
2.2	Service Provider	2
2.3	Cloud data center	1

2.4	Collaboration	2
2.5	Service oriented architecture	2
2.6	Data center Based Service Oriented Architecture	1
3	**VIRTUALIZATION (10)**	
3.1	Components of Virtualization	1
3.2	OS Level Virtualization	1
3.3	Application Virtualization	1
3.4	Desktop Virtualization	1
3.5	Network Virtualization	1
3.6	Virtualization ROI	1
3.7	Benefits of Virtualization	1
3.8	Server Virtualization	1
3.9	Tools for Virtualization	2
4	**SECURITY IN THE CLOUD (10)**	
4.1	Cloud security challenges	1
4.2	Security concerns in cloud computing	1
4.3	Security requirements for the architecture	1
4.4	Security patterns and Architectural elements	1
4.5	Key strategies for secure operation	1
4.6	Multitenancy	1
4.7	Data encryption Application and limits	1
4.8	Cloud data security and sensitive data categorization	2
4.9	Security controls	1
5	**STORAGE IN THE CLOUD(8)**	
5.1	Server and storage I/O fundamentals	1
5.2	Storage Reliability, Availability and Serviceability (RAS)	1
5.3	Virtual, Physical and Cloud data protection	1
5.4	Data Footprint Reduction techniques	2
5.5	Storage services and functionalities	1
5.6	Storage system architectures	1
5.7	Storage virtualization	1
	Total	**50**

Course Designer:

1. Ms. G. MadhuPriya gmadhupriya@tce.edu

Sub Code	Lectures	Tutorial	Practical	Credit
CTQ	4	0	-	4

CTQ Multicore Processor and Systems 4:0

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 aims at facilitating the student to understand the Multicore architectures and address the issues related to optimization and performance for the multicore processors.

Competencies

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

1. Identify different Multicore architectural models.
2. Understand the Optimizing Memory Transactions techniques for multicore systems
3. Apply the Cross core interference techniques on multicore processors.
4. Understand the runtime optimization methods for multicore system
5. Identify the programming issues in multicore programming model

Assessment Pattern

	Bloom's Category	Test 1	Test 2	End-semester examination
1	Remember	20	20	10
2	Understand	40	30	40
3	Apply	40	50	50
4	Analyze	0	0	0
5	Evaluate	0	0	0

Course Level Learning Objectives

Remember

1. Define Multicore Processor.
2. List out the types of Multicore Architecture.
3. Define Cell broadband engine.

4. Define Hotspot.
5. List out types of programming model?

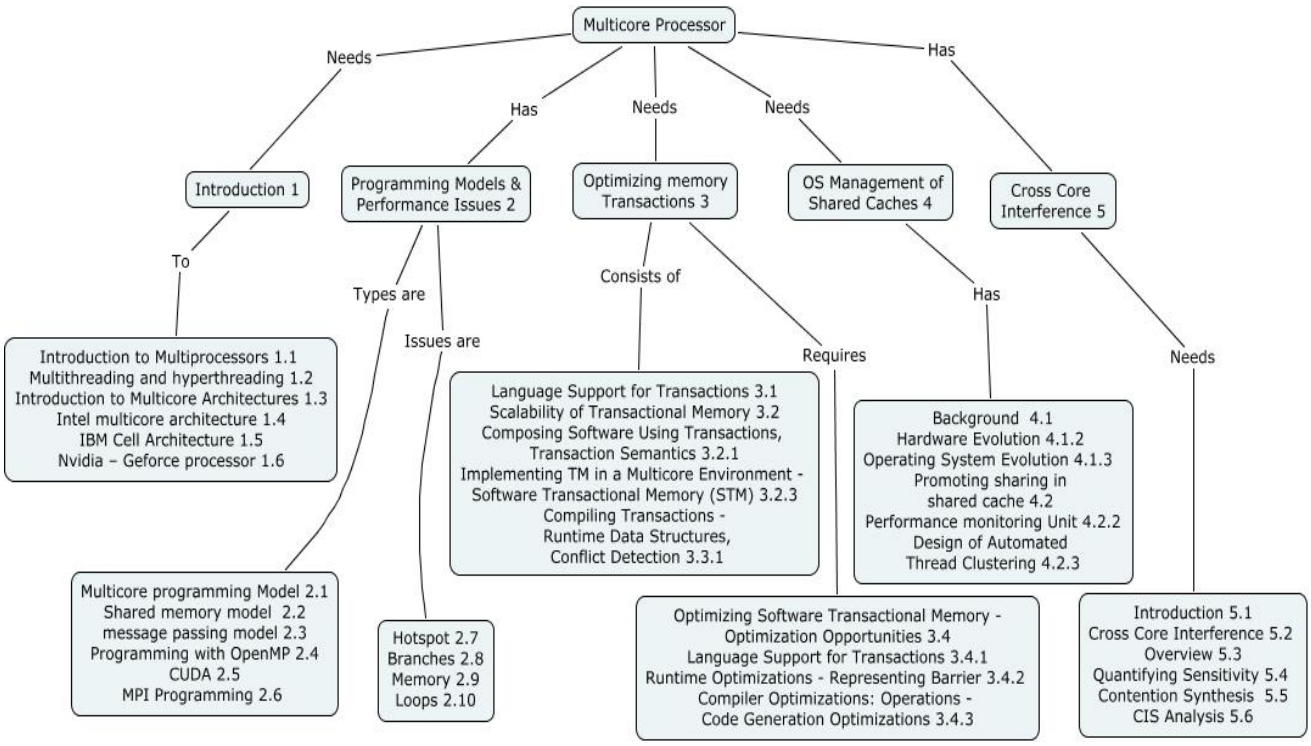
Understand

1. Explain Multithreading.
2. Summarize the significance of Shared Cache Memory.
3. Compare and contrast IBM and Intel multicore processors.
4. Discuss the importance of Implementing TM in a Multicore Environment.
5. Review CUDA programming model.

Apply

1. Compare the significance of Shared memory model and Message passing model.
2. Diagnose the significance of Branching in multiprocessor.
3. Illustrate the significance of Compiler Optimizations in optimized memory transaction.
4. Construct a shared memory program for illustrating the hash table indexing of data.
5. Compare and contrast Hardware Evolution and Operating System Evolution.
6. Illustrate the significance of Cross Core Interference.

Concept map



Syllabus

Introduction - Introduction to Multiprocessors – Multithreading – Hyper threading – Introduction to Multicore Architectures - Intel Core micro architecture,– IBM Cell Architecture, Nvidia – Geforce processor. **Multicore Programming Models and Performance Issues** Multicore programming Model – Shared memory model, message passing model, Programming with OpenMP, CUDA - MPI Programming, Hotspot, Branches, Memory, Loops. **Optimizing Memory Transactions For Multicore Systems** Introduction, Multicore Programming with Transactions - Language Support for Transactions - Scalability of Transactional Memory - Composing Software Using Transactions - Transaction Semantics, Implementing TM in a Multicore Environment, Software Transactional Memory (STM) - Compiling Transactions - Runtime Data Structures - Conflict Detection, Optimizing Software Transactional Memory - Optimization Opportunities - Runtime Optimizations - Compiler Optimizations: Representing Barrier Operations - Code Generation Optimizations. **Operating System Management Of Shared Caches On Multicore Processors** Background – Hardware Evolution – Operating System Evolution, Promoting sharing in the shared cache – Introduction – Performance monitoring Unit – Design of Automated Thread Clustering Mechanism. **Cross Core Interference – Direct And In Direct Analysis** Introduction – Cross Core Interference – Overview – Quantifying Sensitivity – Contention Synthesis – CIS Analysis.

References

1. Stephen W. Keckler, Kunle Olukotun, H. Peter Hofstee "Multicore Processors and Systems (Integrated Circuits and Systems)" Springer New York Dordrecht Heidelberg London-2009, ISBN: 1441902627.
2. Richard Gerber, Aart j.c. Bik, Kevin B. Smith, and Xinmin Tian" The Software Optimization Cookbook" Second Edition, Intel Press-2006.
3. Shameem Akhter, Jason Roberts, "Multi - Core Programming – Increase Performance through software multi-threading**"** Intel Press, April 2006.
4. http://software.intel.com/en-us/articles/multi-core-processor-architecture-explained/
5. http://domino.research.ibm.com/comm/research.nsf/pages/r.arch.innovation.html
6. Mars, Jason and Tang, Lingjia and Soffa, Mary Lou, "Directly characterizing cross core interference through contention synthesis", Proceedings of the 6th International Conference on High Performance and Embedded Architectures and Compilers, 2011, ACM - 978-1-4503-0241-8, pages 167—176.
7. David Tom, "Operating System Management of Shared Caches on Multicore Processors" 2010, Thesis, URL - http://www.eecg.toronto.edu.

Course Contents & Lecture Schedule

No.	Topic	No. of Lectures
1	**Introduction To Multiprocessors (10)**	
1.1	Introduction to Multiprocessors	1
1.2	Multithreading and hyperthreading	1
1.3	Introduction to Multicore Architectures	2
1.4	Intel multicore architecture	2
1.5	IBM Cell Architecture	2
1.6	Nvidia – Geforce processor.	2
2	**Multicore Programming Models And Performance Issues (13)**	
2.1	Parallel programming Models	1
2.2	Shared memory model	1
2.3	message passing model	1
2.4	Programming with OpenMP	2
2.5	CUDA	2
2.7	Hotspot	1
2.8	Branches	2
2.9	Memory	1
2.10	Loops	2
3	**Optimizing Memory Transactions For Multicore Systems (10)**	
3.1	Language Support for Transactions	1
3.2	Scalability of Transactional Memory	1
3.2.1	Composing Software Using Transactions, Transaction Semantics	1
3.2.3	Implementing TM in a Multicore Environment - Software Transactional Memory (STM)	1
3.3.1	Compiling Transactions - Runtime Data Structures, Conflict Detection	1
3.4	Optimizing Software Transactional Memory - Optimization Opportunities	1
3.4.1	Language Support for Transactions	1

No.	Topic	No. of Lectures
3.4.2	Runtime Optimizations - Representing Barrier	1
3.4.3	Compiler Optimizations: Operations - Code Generation Optimizations	2
4	**Operating System Management Of Shared Caches On Multicore Processors (9)**	
4.1	Background	1
4.1.2	Hardware Evolution	1
4.1.3	Operating System Evolution	1
4.2	Promoting sharing in the shared cache	2
4.2.2	Introduction, Performance monitoring Unit	2
4.2.3	Design of Automated Thread Clustering	2
5	**Cross Core Interference – Direct And In Direct Analysis (8)**	
5.1	Introduction	1
5.2	Cross Core Interference	1
5.4	Quantifying Sensitivity	2
5.5	Contention Synthesis	2
5.6	CIS Analysis	2
	Total Hours	50

COURSE DESIGNER

P.Chitra, pccse@tce.edu