GALGOTIAS UNIVERSITY

Syllabus of

Course Book M.Tech.(CSE) 2015-16

Name of School: Computing Science and Engineering

Department: Computer Science and Engineering

Year: 2015-16

Curriculum

Sl. No	Course Code	Name of the Course	L	T	P	C
Semester - I						
1	MAT502	Advanced Numerical & Statistical Methods	3	0	0	3
2	CSE501	Advanced Design and Analysis of Algorithms	3	0	0	3
3	CSE503	Advanced Operating Systems	3	0	0	3
4	CSE504	Advanced Computer Networks	3	0	0	3
5	CSE511	Knowledge Based System Design	3	0	0	3
6	ENG584	Professional English-1	0	0	4	2
7	CSE551	Advanced Design and Analysis of Algorithm Lab	0	0	2	1
8	CSE552	Advanced Computer Network Lab(NS-2 in Linux Environment)	0	0	2	1
9	CSE553	KBSD Lab Using MAT LAB	0	0	2	1
Semester - II						
1	LLL522	Quantitative and Communication Proficiency	0	0	4	2
2	CSE524	Network Security	3	0	0	3
3	CSE507	Advanced Computer Graphics	3	0	0	3
4	CSE512	Business Analytics	3	0	0	3
5	CSE509	Program Elective-1	3	0	0	3
6	CSE522	Program Elective-2	3	0	0	3
Semester - III						
1	CSE613	Requirement Analysis and Project Management	3	0	0	3
2		Program Elective-3	3	0	0	3
3		Program Elective-4	3	0	0	3
4	CSE651	Requirement Analysis and Project Management Lab	0	0	2	1
5	CSE652	M. Tech Dissertation Part-1	0	0	0	5
Semester - IV						
1	CSE650	M. Tech Dissertation-Final	0	0	30	15

List of Electives

Program Elective- I

Sl No	Course Code	Name of the Electives	L	T	P	C
1	CSE508	Coding and Information Theory	3	0	0	3
2	CSE509	High Performance Computer Systems	3	0	0	3
3	CSE510	Parallel Algorithms	3	0	0	3
4	CSE517	Distributed Systems	3	0	0	3

Program Elective-II

Sl No	Course Code	Name of the Electives	L	T	P	C
1	CSE519	Advances in Compiler Design	3	0	0	3
2	CSE514	Data Compression	3	0	0	3
3	CSE515	Natural Language Processing	3	0	0	3
4	CSE522	Mobile Computing	3	0	0	3

Program Elective-III

Sl No	Course Code	Name of the Electives	L	T	P	C
1	CSE518	Interaction Design	3	0	0	3
2	CSE554	Data Compression Lab	3	0	0	3
3	CSE561	Real Time System	3	0	0	3
4	CSE555	Web Services Lab	3	0	0	3

Program Elective-IV

Sl No	Course Code	Name of the Electives	L	T	P	C
1	CSE523	Big Data Analytics	3	0	0	3
2	CSE513	Transaction Processing	3	0	0	3
3	CSE516	Web Services	3	0	0	3
4	CSE511	Grid and Cluster Computing	3	0	0	3

Detailed Syllabus

Course Code: MAT502	Advanced Numerical and Statistical Methods	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Course Objectives

1. The main objective of this course is to understand and implement various concepts of numerical analysis and statistics to solve real life problems.
2. Analysis of Statistical Data: Frequency distribution; Frequency curve and histogram; Measure of central tendency and dispersion.

Course Outcomes

At the end of the course, students will be able to:

1. Understand the various approaches dealing the data using theory of probability.
2. Analyze the different samples of data at different level of significance using various hypothesis testing.
3. Develop a framework for estimating and predicting the different sample of data for handling the uncertainties.
4. Understand error, source of error and its affect on any numerical computation and also analyzing the efficiency of any numerical algorithm.
5. Learn how to obtain numerical solution of nonlinear equations using Bisection, Newton –Raphson and fixed-point iteration methods Understand and apply linear programming concepts to real time applications

Module -I

System of Linear Equations: Direct Methods- Gauss elimination – Pivoting, Partial and Total Pivoting, Triangular factorization method using Crout LU decomposition, Cholesky method, Iterative Method- Gauss- Seidel and Jacobi method, ill conditioned matrixSystem of Non-linear equation- Newton Raphson and Modified Newton Raphson Method. Iterative methods.

Module -II

Interpolation and Approximation: Lagrange, Spline and Hermite interpolation, Approximations, Error of approximation, Norms for discrete and continuous data, Least square approximation.

Module -III

Numerical Integration: Newton Cotes closed Quadrature, Gauss Legendre Quadrature, Multiple Integration.

Module -IV

Numerical Solution of Differential Equations: Finite Difference Schemes, Numerical

solution of Ordinary differential equation using Modified Euler's method, Runge-Kutta method of 2nd, 3rd and 4th orders, Predictor- Corrector method, Solution of Laplace's and Poisson's equations by Liebman's method, Solution of one dimensional time dependent heat flow.

Module-V

Probability and statistics: Review of concept of probability, Random Variables, Continuous and discrete distribution function, moments and moments generating functions, Binomial, Poisson, Negative Binomial, Geometric and Hyper-geometric Distributions, Uniform, Normal, Exponential, Gamma and Beta distributions. Point and Interval estimation, Testing of Hypothesis (t-test and chi square test), Analysis of variance and Introduction of Design of experiments.

Text Books:

1. Numerical Methods for Scientific and Engineering Computation (6^{th}edition) by Jain, Iyengar & Jain, New Age International publishers.
2. Probability & Statistics for Engineers & Scientists (9^{th}edition) by R.E.Walpole, R,H,Myers&K.Ye.

Reference Books:

1. Numerical Methods by E Balagurusamy, Tata McGraw Hill
2. Curtis F. Gerald and Patrick O Wheatley, Applied Numerical Analysis, Pearson Education Ltd.
3. Introductory Methods of Numerical Analysis by S.S. Sastry, PHI learning Pvt Ltd.

Course Code: CSE501	Advanced Design and Analysis of Algorithms	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	3	0	0	3
Prerequisite/Exposure	Data Structures and Algorithms				
Co-requisites					

Course Objectives

1. To know the importance of the complexity of a given algorithm.
2. To study various algorithmic design techniques.
3. To utilize data structures and/or algorithmic design techniques in solving new problems.
4. To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.

Course Outcomes

At the end of the course, students will be able to:

1. Analyze the complexity of the algorithms and use technique divide and conquer to solve the problems
2. Identify feasible solutions for different problems through greedy method and minimize the solutions space and to solve the problems through dynamic programming.
3. Solve the problems through graph algorithms.
4. Justify that a certain problem is NP-Complete
5. Understand and apply linear programming concepts to real time applications.

Course Content

Module I: Introduction

Overview of algorithmic design**,** asymptotic notation and its properties, Growth of Functions, Time complexity and Analysis of algorithms, Recurrence Relations.

Module II: Sorting and Searching Algorithms

Brute Force Method - Sorting in Quadratic time, insertion , selection and Bubble sort; Divide and conquer method-Sorting in Logarithmic time – Quick Sort , merge Sort, Shell Sort , Heap sort; Non-comparison sorts - Sorting in Linear Time - Counting Sort, Radix Sort , Bucket Sort; Worst Case and best case analysis of all sorting algorithms; Linear Search, Binary Search, Hashing, Randomized select , randomized quick sort.

Module III: Algorithms for Trees

Binary Tree - Binary Tree traversals, Binary Search Tree, heap, priority Queues, Red Black Trees, B-Trees.

ModuleIV:Graph Algorithms

Graph Searching- Breadth-First Search, Depth-First Search, DAGs and topological sorting, minimum spanning tree, shortest path, backtracking, Network flow algorithms.

Module V: Greedy Algorithms, Amortized Analysis and Dynamic Programming

Longest common subsequence, Greedy Algorithms - Knapsack problem; Huffman codes, Algorithms for String Matching, Theory of NP-completeness; Turing machines and the halting problem, Applications of Algorithms in Databases, Information Retrieval and Web Searching, Data Mining.

Text Books

1. Cormen, Leiserson, Rivest and Stein, "Introduction to Algorithms", 2nd Edition, by, McGraw-Hill, 2000.
2. E. Horowitz, and S. Sahni, "Fundamentals of Computer Algorithms", Computer Science Press (1978).

Reference Books

1. Jon Kleinberg and Eva Tardos. Algorithm Design. Pearson Education, 2007.
2. Sanjoy Das Gupta, Christos Papadimitriou, Umesh Vazirani, Algorithms 1st Edition, Mcgraw Higher Ed, 2006.
3. Alfred V. Aho, John E. Hopcroft, Jeffery D.Ulman, Data Structures and Algorithms, Pearson; 1st edition, 2001.

Course Code: CSE501	Advanced Operating Systems	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	3	0	0	3
Prerequisite/Exposure	Data Structures and Algorithms				
Co-requisites					

Course Objectives:

1. To provide the fundamental principles of modern operating systems that explores design aspects of modern operating systems.
2. To utilize data structures and/or algorithmic design techniques in solving new problems.
3. To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.

Course Outcomes:

1. On completion of this course the student should be able to understand and evaluate operating system implementations.
2. Develop system software modules, Write and debug concurrent programs,
3. Debug complex systems and low-level software and Work with distributed and real time OS.

Course Content

Module I: Introduction

Operating system concept - processes and threads, process model, process creation, process termination, process hierarchies, and process states, Implementation of processes, Threads- Thread model, thread usage, Implementation of threads in user space and kernel, Hybrid implementations.

Module II: Inter proses Communication

Race conditions, critical regions, Mutual Exclusion with busy waiting, sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing; Scheduling- scheduling in batch systems, Interactive systems, Real time systems, Thread scheduling.

Module III: Deadlocks

Deadlocks-Introduction, Deadlock Detection and Recovery – Deadlock Detection with one resource of each type, with multiple resource of each type, recovery from deadlock; Deadlock Avoidance, Deadlock Prevention

Module IV: Memory and Device Management

Introduction, Swapping, Paging, Virtual memory – Demand paging, page replacement Algorithms; File System Management- Organization of File System, File Permissions, MS DOS and UNIX file system case studies, NTFS; Device Management- I/O Channels, Interrupts and Interrupt Handling, Types of device allocation.

Module V: Distributed Operating Systems

Distributed operating system concept – Architectures of Distributed Systems, Distributed Mutual Exclusion, Distributed Deadlock detection, Agreement protocols, Threads, processor Allocation, Allocation algorithms , Distributed File system design; Real Time Operating

Systems: Introduction to Real Time Operating Systems, Concepts of scheduling , Real time Memory Management

Reference Books

1. Mukesh Singhal and Niranjan, “Advanced Concepts in Operating Systems”, TMH, 1st Edition, 2001
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Pearson Education, 2nd Edition, 2006
3. Andrew S. Tanenbaum, “Distributed Operating Systems”, Pearson Education, 2nd Edition, 2001
4. Pradeep K. Sinha, “Distributed Operating Systems and concepts”, PHI, First Edition, 2002
5. Mode of Evaluation Quiz/Assignment/ Seminar/Written Examination

Course Code: CSE504	Advanced Computer Networks	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	3	0	0	3
Prerequisite/Exposure	Data Structures and Algorithms				
Co-requisites					

Course Objectives:

1. To go beyond the basic level of understanding that is typically offered at an undergraduate networking course.
2. To utilize data structures and/or algorithmic design techniques in solving new problems.
3. To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.

Course Outcomes:

1. On completion of course students will be able to understand the fundamental concepts
2. Develop the routing tables and addressing, transport protocols.
3. Student design the congestion control, emerging distributed applications, and wireless networking.

Course Content

Module I: Networking Standards and Specification

Networking standards and specifications, Need for standardization, ISO and the IEEE standards, The IEEE 802 Project

Module II: Addressing and Routing

Network names and addresses, Physical layer addressing: the MAC address, Network layer addressing: The IP address, Network layer address: The IPX address.

Module III: Overview of OSI and TCP/IP Protocol Suite

Converting network names to IP addresses, Resolving IP addresses to physical addresses, Addressing and routing

Module IV: TCP/IP Protocol Suite

TCP/IP Protocol Suite, TCP/IP Protocol Suite advantages, Internet Protocol (IP), Transport Layer Protocols -TCP and UDP, File Transfer protocols - FTP and TFTP, Mail and news protocols - SMTP, POP3, NNTP and IMAP, Other Protocols Suite – ICMP and ARP.

Module V: Other Networking Protocols

The IPX/SPX Protocol Suite, NetBEUI, AppleTalk Protocol, File sharing protocols - SMB, NCP, and NFS, Routing protocols - RIP, OSPF and BGP, Network Management Protocol – SNMP and CIMP, Convergent Protocols – H.323 and SIP

Reference Books

1. Behrouz A. Forouzan, TCP/IP Protocol Suite, Third Edition, Tata McGraw-Hill, 2005.
2. W. Richard Stevens, TCP/IP Illustrated, The Protocols, Pearson Education, 2004.
3. D. E. Comer, Internetworking with TCP/IP Principles, Protocols and Architecture Vol - I, Pearson Education, 2001.

Course Code: CSE511	Knowledge Based System Design	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. Student acquires the theoretical fundaments and the practical abilities to be capable to develop a knowledge-based system.
2. Student will be able to compile and represent a problem´s knowledge, and to develop or adapt existent motors to reason about problems and solve them.
3. Students should able to develop software, and the most extended implementations based in inference motors.
4. Understand the roll that the knowledge-based systems have in the society

Course Outcomes:

1. To get the concepts of Prolog.
2. To understand the knowledge-based systems representation.
3. To implement a small knowledge- based system.
4. To understand the role of Artificial Intelligence, Expert Systems and Decision Models in managerial decision-making.
5. To get an In-Depth Knowledge of Machine Learning.

Course Content

Module 1 (Introduction): Introduction to Programming Logic Introduction to Logic, Propositional Logic concepts, Logic Programming in Prolog.

Module 2 (Knowledge Engineering Knowledge Engineering): The human expert and an Artificial Expert, knowledge base and inference engine, knowledge acquisition and knowledge representation.

Module 3 (Problem Solving): Knowledge Engineering Knowledge Engineering: The human expert and an Artificial Expert.

Module 4 (Expert Systems): Tools for building expert systems, case based reasoning, semantic of expert systems, modeling of uncertain reasoning, applications of semiotic theory; Designing for explanation.

Module 5 (Expert System Architecture and Programming): Expert system architectures, high level programming languages.

Text Books:

1. PROLOG: Programming for Artificial Intelligence, 3e, by Ivan Bratko.
2. Introduction to Expert Systems, 3rd Edition by Pearson Education 2007 by Peter Jackson.
3. AI and Expert Systems: a comprehensive guide, C language, 2nd edition, McGraw-Hill 1990 by Robert I. Levine, Diane E. Drang, Barry Edelson.

Reference Books:

1. Artificial Intelligence and Intelligent Systems, 4th impression , Oxford University Press, 2007 by Padhy N.P.

2. Expert Systems: Theory and Practice, 4th printing, PrenticeHall of India, 2001, by Jean-Louis Ermine.

3. Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, 2007 by Stuart Russell, Peter Norvig.

Course Code: ENG 584	Professional English-I	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	4	2
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. To develop the professional and communication skills of learners in a technical environment.
2. To enable the students to acquire functional and technical writing skills.
3. To acquire state-of-the-art presentation skills in order to present technical topics to both technical and non-technical audience.

Course Outcomes:

1. The learners will be able to exhibit their language proficiency
2. The learners should develop the skill in *Describing, Investigating, Designing and Making* and *Using Technology.*
3. students should acquire functional and technical writing skills.
4. To acquire state-of-the-art presentation skills in order to present technical topics to both technical and non-technical audience

Course Content

Module-I: Functional Language

Basic structures- Tense agreement, Prepositional phrases - Techno-words : Basic Concepts 62, 63 - Pronunciation : sounds of syllables: Past tense & plural endings -Technical Expression Organisational techniques in technical writing -Guided writing: Paragraph Writing, Note Making -Presentation Skills Techniques of presentation (general topics: speech without visual aids) -Listening to speeches and comprehending -Graphical Skills Flow chart : Process and Functional description

Module-II: Functional Language

Basic structures- Voice, Conditionals -Techno-words : Basic Concepts 64,65,67 - Pronunciation : Word Stress: two syllable words -Technical Expression Mechanics of Technical Writing and Syntax - Guided writing: Letter and email -Presentation Skills Interpersonal Communication Skills -Writing techniques for Power point presentation, Group Discussion -Graphical Skills Technical Illustrations and Instructions

Module-III

Functional Language Basic structures- Modal Verbs and Phrasal verbs -Techno-words: Basic Concepts 68,69,70,71 -Pronunciation: Word Stress: compound words -Technical Expression Mechanics of Technical Writing and Syntax -Guided writing: Technical Description

1. English Vocabulary in Use Advanced, McCarthy & Felicity, CUP, 2003
2. Sky Pronunciation CD-ROM
3. Cambridge Advanced Learner's Dictionary CD-ROM
4. English Master : Grammar

References

1. Writing, Researching, Communicating, Keith et al, Tata McGraw-Hill, 1989
2. Advanced English Grammar, Martin, CUP, 2006

Course Code: CSE551	Advanced Design and Analysis of Algorithms Lab	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	2	1
Prerequisite/Exposure	Data Structures and Algorithms				
Co-requisites					

List of Experiments:

1. Write C++ programs to implement the following: a) Prim's algorithm. b) Kruskal's algorithm.
2. Write a C++ program to find optimal ordering of matrix multiplication. (Note: Use Dynamic programming method).
3. Consider the problem of eight queens on an (8x8) chessboard. Two queens are said to attack each other if they are on the same row, column, or diagonal. Write a C++ program that implements backtracking algorithm to solve the problem i.e. place eight non-attacking queens on the board.
4. Write a C++ program to find the strongly connected components in a digraph.
5. Write a C++ program to implement file compression (and un-compression) using Huffman's algorithm. .
6. Write a C++ program to implement dynamic programming algorithm to solve all pairs shortest path problem.
7. Write a C++ program to solve 0/1 knapsack problem using the following: a) Greedy algorithm. b) Dynamic programming algorithm. c) Backtracking algorithm. d) Branch and bound algorithm.
8. Write a C++ program that uses dynamic programming algorithm to solve the optimal binary search tree problem.
9. Write a C++ program for solving traveling sales persons problem using the following: a) Dynamic programming algorithm. b) The back tracking algorithm.

Course Code: CSE552	Advanced Computer Networks Lab	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	2	1
Prerequisite/Exposure	Computer Networks				
Co-requisites					

List of Experiments:

1. Configuration and logging to a CISCO Router and introduction to the basic user Interfaces. Introduction to the basic router configuration and basic commands.
2. Configuration of IP addressing for a given scenario for a given set of topologies.
3. Configure a DHCP Server to serve contiguous IP addresses to a pool of four IP devices with a default gateway and a default DNS address. Integrate the DHCP server with a BOOTP demon to automatically serve Windows and Linux OS Binaries based on client MAC address.
4. Configure, implement and debug the following: Use open source tools for debugging and diagnostics. a. ARP/RARP protocols b. RIP routing protocols c. BGP routing d. OSPF routing protocols e. Static routes (check using netstat)
5. Configure DNS: Make a caching DNS client, and a DNS Proxy; implement reverse DNS and forward DNS, using TCP dump/Wireshark characterize traffic when the DNS server is up and when it is down.
6. Configure FTP Server on a Linux/Windows machine using a FTP client/SFTP client characterize file transfer rate for a cluster of small files 100k each and a video file of 700mb.Use a TFTP client and repeat the experiment.
7. Configure a mail server for IMAP/POP protocols and write a simple SMTP client in C/C++/Java client to send and receive mails
8. Implement Open NMS+ SNMPD for checking Device status of devices in community MIB of a linux PC. Using yellow pages and NIS/NFS protocols implement Network Attached Storage Controller (NAS). Extend this to serve a windows client using SMB. Characterise the NAS traffic using wireshark.

Course Code: CSE553	KBSD Lab using MATLAB	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	2	1
Prerequisite/Exposure					
Co-requisites					

CSE 553 KBSD Lab using MATLAB

List of Experiments

a. Study of Prolog.
b. Write a program in prolog for first order predicates for person activity system.
3. Write a first order predicate logic in prolog for cars.
4. Write a program of first predicate logic in prolog for a family relation problem.
5. Write a program for factorial in prolog.
6. Write a program to implement Towers of Hanoi.
7. Write a program for menu driven program for member concatenation, permutation, add and delete function in prolog.
8. Write a program in prolog to find the union and intersection of two given list.
9. Write a program to implement Travelling Salesman Problem.
10. Write a program to implement 8 puzzle problem.
11. Write a program to implement water jug problem.
12. Write a program to solve monkey banana problem.

Experiments based on advanced topics:

13. Write a program to implement breadth first search, depth first search and best first search.
14. Write a program to solve traversal problem using mean end analysis.

Semester – II

Course Code: LLL 522	**Quantitative and Communication Proficiency**	L	T	P	C
Version No.1.0	**Date of Approval: 19/06/2015**				
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. This module would train the students on the quick ways to solve quantitative aptitude problems.
2. To equip the students with the required soft skills that would instill confidence and courage in them, to take up new opportunities for their career.

Course Outcome:

The students will gain the ability to solve quantitative aptitude problems in a simple way using short-cut methods, within a short time span given during the placement drives.

Course Content

Quantitative Aptitude

Number System, Partnership, Compound Interest, Simple Interest, Profit and Loss, Problems on Clock, Calendar and Cubes, Permutation and Combination, Allegation and mixtures, Time and Distance, Height and Distance, Problems on Ages, Trains, Boats and Streams, Probability.

Communication Proficiency

Self analysis to challenges -Attitude- perceptions– Positive approach – ideas & approach

Goal setting – vision - Time management - planning -Entrepreneurial skills - Leadership skills

People management – team work, leadership -Decision making – problem identification

Interview skills – getting familiar with one’s CV – presentation and performance - giving and receiving feedback, setting expectations and exhibiting professional behavior.

Course Code: CSE 524	Network Security	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015				
Prerequisite/Exposure					
Co-requisites					

Course Objectives

1. To understand the fundamentals of network security.
2. To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
3. To understand the various key distribution and management schemes.
4. To understand how to deploy encryption techniques to secure data in transit across data networks.
5. To design security applications in the field of Information technology.

Course Outcomes

At the end of the course, students will be able to:

1. Compare various Security Techniques Design Secure applications Inject secure coding in the developed applications.
2. Implement basic security algorithms required by any computing system.
3. Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
4. Analyze the possible security attacks in complex real time systems and their effective countermeasures.
5. Identify the security issues in the network and resolve it.
6. Evaluate security mechanisms using rigorous approaches, including theoretical derivation, modeling, and simulations.
7. Formulate research problems in the computer security field.

Unit I:INTRODUCTION 10 lecture hours

Services, Mechanisms and attacks-Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography).FINITE FIELDS AND NUMBER THEORY: Groups, Rings, Fields-Modular arithmetic-Euclid's algorithm Finite fields- Polynomial Arithmetic –Prime numbers-Fermat's and Euler's theorem-Testing for primality -The Chinese remainder theorem- Discrete logarithms.

Unit II: BLOCK CIPHERS & PUBLIC KEY ENCRYPTION 10 lecture hours

Data Encryption Standard-Block cipher design principles-block cipher modes of operation Advanced Encryption Standard (AES)-Triple DES-Blowfish-RC5 algorithm. Public key encryption: Principles of public key cryptosystems-The RSA algorithm.

Unit III: HASH FUNCTIONS AND DIGITAL SIGNATURES 10 lecture hours

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function and MAC –MD5 - SHA - HMAC – CMAC - Digital signature and authentication protocols – DSS – EI Gamal – Schnorr.

Unit IV : E-MAIL, IP & WEB SECURITY 8 lecture hours

E-mail Security: Pretty Good Privacy-S/MIME. IPSecurity: Overview of IPSec - IP security policy-Encapsulation Security Payload (ESP)-Combining Security Associations-Internet Key Exchange.

Unit V: SYSTEM SECURITY **9 lecture hours**

Authentication applications – Kerberos – X.509 Authentication services - Firewalls – Types of Firewalls- Firewall design principles- Trusted System.

Text Books

1. Behrouz A. Ferouzan, ―Cryptography & Network Security‖, Tata Mc Graw Hill, 2007.
2. Bruce Schneier and Neils Ferguson, ―Practical Cryptography‖, First Edition, Wiley Dreamtech India Pvt Ltd, 2003.
3. Charles Pfleeger, ―Security in Computing‖, 4th Edition, Prentice Hall of India, 2006.
4. Charlie Kaufman and Radia Perlman, Mike Speciner, ―Network Security, Second Edition, Private Communication in Public World‖, PHI 2002.

Reference Books

1. Douglas R Simson ―Cryptography – Theory and practice, First Edition, CRC Press, 1995.
2. Man Young Rhee, ―Internet Security: Cryptographic Principles‖, ―Algorithms and Protocols‖, Wiley Publications, 2003.
3. Ulysess Black, ―Internet Security Protocols‖, Pearson Education Asia, 2000.
4. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.

Course Code: CSE507	Advanced Computer Graphics	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	2	1
Prerequisite/Exposure					
Co-requisites					

Course Objectives

1. To study the graphics techniques, packages and algorithms.
2. To enable the Students to understand the Graphics rendering and hardware.
3. To enable the Students to learn visualization techniques.

Course Outcomes

At the end of the course, students will be able to:

1. Create interactive graphics applications using one or more graphics application programming interfaces.
2. Use illumination models.
3. Explain graphics hardware.
4. Apply surface rendering and visualization techniques.

Course Content

Unit I:INTRODUCTION **8 lecture hours**

Overview, Modeling, Procedural Models, Fractal Models, and Grammar based models, particle systems, and viewing, Rasterization and Ray tracing

Unit II: Illumination **8 lecture hours**

Vertex/Geometry/Pixel programming, Illumination mode, specular reflection model, shading models for curve surfaces, Radiosity method, Rendering, Recursive ray tracing, Texture mapping.

Unit III: Graphics Hardware **8 lecture hours**

Graphics hardware architecture, Object representation and levels of detail.

Unit IV :Surface Rendering **8 lecture hours**

Parametric and implicit surfaces, Meshing, Visibility and shadow computation, Global illumination.

Unit V: Visualization Techniques **9 lecture hours**

Introduction to volume visualization, Introduction to animation, Image based rendering, Filler

Text Books

1. Watt A. and M. Watt, Advanced, Animation and Rendering Techniques, Addison Wesley, 1992.
2. Hearn D. and P. Baker, Computer Graphics C Version, Pearson Education India; 2 edition, 2002.

Reference Books

1. Neider, J., T. Davis, and M. Woo, OpenGL Programming Guide, Addison-Wesley, 1993.

2. Luebke D., M. Reddy, J. Cohen, A. Varshney, B. Watson, R. Huebner, Level of Detail for 3D Graphics, 2003.
3. James D. Foley, Andries van Dam, Steven K. Feiner and John Hughes, Computer Graphics: Principles and Practice, Second Edition in C, Addison-Wesley, 1995.
4. Dan Ginsburg, Budi Purnomo, Dave Shreiner and Aatab Munshi, OpenGL ES 3.0 Programming Guide 2nd Edition, Kindle Edition, 2014.

Course Code: CSE 512	Business analytics	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015	0	0	2	1
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. To gain an understanding of how managers use business analytics to formulate and solve business problems and to support managerial decision making.

2. To become familiar with the processes needed to develop, report, and analyze business data.

3. To learn how to use and apply Excel and Excel add-ins to solve business problems.

Course Outcomes:

1. Enable all participants to recognise, understand and apply the language, theory and models of the field of business analytics
2. Foster an ability to critically analyse, synthesise and solve complex unstructured business problems
3. Encourage an aptitude for business improvement, innovation and entrepreneurial action
4. Encourage the sharing of experiences to enhance the benefits of collaborative learning

Course Content

Module I: Overview of Business Analytics

Introduction to Analytics, Davenport article - "Competing on Analytics" , LaValle et al. article - "Analytics: The New Path to Value"

Module II: Visualization/ Data issues

Organization/sources of data, Importance of data quality, Dealing with missing or incomplete data, Data Classification, Davenport and Harris article - "The Dark Side of Customer Analytics"

Module III: Introduction to Data Mining

Introduction to Data Mining, Data Mining Process, Data mining tool XLMiner, Loveman article – "Diamonds in the Data Mine", Market Basket Analysis – Shmueli Chapter 13, Classification and Regression Trees – Shmueli Chapter 14

Module IV: Introduction to Decision Modelling

Optimization Use of Excel to solve business problems: e.g. marketing mix, capital budgeting, portfolio optimization • Decision Making under

Module V: Types of problems, inventory management, capital investment analysis, market share estimation, sensitivity analysis

Text Books:

1. Business Analytics 2nd Edition, James Evans, Person
2. Business Analytics: The Science of Data - Driven Decision Making, U. Dinesh Kumar Wiley

Reference Books:

1. Business Analytics: Data Analysis &Decision Making 6th Edition, S. Christian Albright (Author), Wayne L. Winston (Author)
2. Essentials of Business Analytics by Jeffrey D Camm.

Semester – III

Course Code: CSE 613	Requirement analysis and Project Managment	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. Students should gain the project management related to managing software development projects.
2. students get familiar with the different activities involved in Software Project Management.
3. Students should obtain the successfully plan and implement a software project management activity, and to complete a specific project in time with the available budget.
4. Identify the different project contexts and suggest an appropriate management strategy.
5. Practice the role of professional ethics in successful software development

Course outcomes:

1. Prepare SRS including the details of requirements engineering.
2. Identify and describe the key phases of project management.
3. Determine an appropriate project management approach through an evaluation of the business context and scope of the project
4. Describe the stages of requirements elicitation.
5. Analyze software requirements gathering.

Course Content

Module 1: Requirements Engineering Overview

Software Requirement Overview – Software Development Roles –Software Development Process Kernels – Commercial Life Cycle Model – Vision Development – Stakeholders Needs & Analysis – Stakeholder needs –Stakeholder activities.

Module 2: Requirements Elicitation

The Process of Requirements Elicitation – Requirements Elicitation Problems – Problems of Scope – Problems of Understanding – Problems of Volatility – Current Elicitation Techniques – Information Gathering – Requirements Expression and Analysis – Validation – An Elicitation Methodology Framework – A Requirements Elicitation Process Model – Methodology over Method – Integration of Techniques – Fact-Finding – Requirements Gathering – Evaluation and Rationalization – Prioritization – Integration and Validation.

Module 3: Requirements Analysis

Identification of Functional and Non Functional Requirements – Identification of Performance Requirements – Identification of safety Requirements – Analysis – Feasibility and Internal Compatibility of System Requirements – Definition of Human Requirements Baseline.

Module 4: Requirements Development

Requirement's analysis – Requirements Documentation – Requirements Development Workflow – Fundamentals of Requirements Development – Requirements Attributes Guidelines Document – Supplementary Specification Document – Use Case Specification Document – Methods for Software Prototyping – Evolutionary prototyping –Throwaway prototyping.

Module 5: Requirements Validation

Validation objectives – Analysis of requirements validation – Activities – Properties – Requirement reviews – Requirements testing – Case tools for requirements engineering.

Text Books:

1. Ian Sommerville, Pete Sawyer, ―Requirements Engineering: A Good Practice Guide‖, Sixth Edition, Pearson Education, 2004.
2. Dean Leffingwe, Don Widrig, ―Managing Software Requirements A Use Case Approach‖, Second Addition, Addison Wesley, 2003.

Reference Books:

1. Karl Eugene Wiegers, ―Software Requirements‖, Word Power Publishers, 2000.
2. Ian Graham, ―Requirements Engineering and Rapid Development‖, Addison Wesley, 1998.
3. Wiegers, Karl, Joy Beatty, ‖Software requirements‖, Pearson Education, 2013.

CSE 651	Requirement Analysis and Project Management LAB	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**				
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. Build a fully functional, interactive, layered, distributed, database-backed software system.
2. Students understand the ground-up as part of a small, agile, development team in a laboratory setting
3. Become acquainted with historical and modern software methodologies
4. Understand the phases of software projects and practice the activities of each phase.

Course Outcomes:

1. Practice clean coding
2. Take part in project management
3. Become adept at such skills as distributed version control, unit testing, integration testing, build management, and deployment

Course Content

OVERVIEW: Definitions, goals, and basic principles of software engineering; differences between software engineering and other fields within computing; engineering vs. craftsmanship; Clean Code.

SOFTWARE PROCESSES: Modeling languages and methodologies, phases, metaphors for software construction; Scrum, Kanban; Setting up a project on GitHub; Basics of Git.

ANALYSIS AND DESIGN: Upstream prerequisites, construction decisions, classes, routines, defensive programming.

OVERVIEW OF UML: Whirlwind tour of UML, including examples from each diagram type.

REVIEW OF COMMON PROGRAMMING LANGUAGES AND TECHNOLOGIES: Python, Java, JavaScript, HTML5, CSS, JSON, Unit testing, Lint tools, IDEs.

ENTERPRISE APPLICATION TECHNOLOGIES: APIs, Relational databases, NoSQL databases, TCP/IP, HTTP, Web applications, Web services; Java stacks (e.g., JavaEE, JAX-RS, Spring, Hibernate), Python stacks (e.g. Flask), Node, Ajax.

APIs: Technologies, REST, business issues, government issues, financial issues, legal issues.

SOFTWARE BEST PRACTICES: Variables, Types, Conditionals, Loops, Disruption, Unusual control structures, Control complexity.

SOFTWARE QUALITY: Measures of quality, improving code, inspections and code reviews, unit testing, integration testing, acceptance testing, code coverage, debugging, refactoring.

PERFORMANCE: Complexity measures, profiling tools, tuning strategies, tuning techniques.

SYSTEM DEPLOYMENT AND MAINTENANCE: Project management, integration, scalability, daily builds, system evolution.

CSE 652	M.Tech Dissertation Part-1	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**				
Prerequisite/Exposure					
Co-requisites					

The Dissertation Work for M.Tech consists of Dissertation Work – I and Dissertation Work–II. Dissertation Work–I is to be undertaken during III semester and Dissertation Work–II, which is generally a continuation of Dissertation Work–I and is to be undertaken during IV semester. At the end of the semester students present the following contents.

• Title

• Abstract

• Introduction

• Literature Survey

• References

Semester- IV

CSE 650	M.Tech Dissertation Final	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**				
Prerequisite/Exposure					
Co-requisites					

At the end of the semester students present the following contents.

- Title
- Abstract
- Introduction
- Literature Survey
- Methodology
- Modules Split-up and Gantt Chart
- Proposed System (Phase 1)
- Equations /Design and software to be used
- Algorithms / Techniques used
- Expected outcomes
- References

Program Elective-1

CSE508	**Coding and Information Theory**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Higher Mathematics, Advanced Computer Networks				
Objectives:	To learn how to formulate and tackle fundamental problems in communications and signal processing through the exposition to 4 main results in information theory.				
Expected Outcome:	On completion of this course the student would be able to deal communication systems.				
Module I	Information Theory Basics				
Entropy, mutual information, chain rules, inequalities, asymptotic equipartition property (AEP), entropy of random processes.					
Module II	Source Coding				
Unique decodability, prefix-free codes, Kraft inequality, AEP-based compression, Huffman coding, arithmetic coding, universal coding.					
Module III	Channel Capacity				
Discrete memory less channels, joint typicality, and achievability & converse proofs of channel capacity theorem, feedback channels, source-channel separation, practical channel codes.					
Module IV	Differential Entropy and Gaussian Channels				
Differential entropy, capacity of AWGN channels, band-limited channels, parallel & fading channels.					
Module V	Rate-Distortion (Lossy Source Coding)				
Quantization, proofs of achievability & converse of rate distortion function					
Reference Books					
1. B.P. Lathi, *An Introduction to Random Signals and Communication Theory*, Intl. 2. S. Haykin, *Communication Systems*, Wiley 3. T. Cover and J. Thomas, Elements of Information Theory , Wiley-Interscience, 1991 4. R.W. Hamming, *Coding and Information Theory*, Prentice Hall					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

Course Code: CSE507	High Performance Computer Systems	L	T	P	C
Version No.1.0	Date of Approval: 19/06/2015				
Prerequisite/Exposure					
Co-requisites					

Course Objectives: To create new technologies that can increase the computational speed accomplishing with accuracy of computing.

Course Outcomes: On completion of this course the students will be able to know about various concepts such as vector processing, out-of-order execution etc., which are used for designing a high performance machine

Course Content

Module I: Modern Computer Architectures

Introduction, High Performance Microprocessors, CISC, Fundamentals of RISC, Second, Generation RISC Processors, Out, of, Order Execution: The Post, RISC Architecture. Memory, Memory Technology, Registers, Caches, Cache Organization, Virtual Memory, Improving Memory Performance.

Module II: Floating, Point Numbers

Reality , Representation , Effects of Floating, Point Representation , Improving Accuracy Using Guard Digits , IEEE Floating, Point Standard , IEEE Storage Format , IEEE Operations Special Values , Exceptions and Traps Compiler Issues.

Module III: Programming and Tuning Software

Optimization Levels , Classical Optimizations , Timing and Profiling , Timing , Subroutine Profiling , Basic Block Profilers , Virtual Memory , Eliminating Clutter , Subroutine Calls , Branches , Branches within Loops.

Module IV: Shared, Memory Parallel Processors

Understanding Parallelism, Shared, Memory Multiprocessors, Symmetric Multiprocessing Hardware Multiprocessor Software Concepts, Techniques for Multithreaded Programs , A Real Example.

Module V: Scalable Parallel Processing

Large, Scale Parallel Computing, Amdahl's Law, Interconnect Technology, A Taxonomy of Parallel Architectures , Shared Uniform Memory MIMD, Shared Non, Uniform Memory MIMD Systems, Distributed, Memory MIMD Architecture , Single Instruction Multiple Data

Text Books:

1. High Performance Computing by Harish G Narula (Author), Monali Shetty (Author)
2. High Performance Cluster Computing - Vol. 1by Buyya

Reference Books

1. Charles Severance, Kevin Dowd, High Performance Computing, Second Edition July 1998.
2. High Performance TCP/IP Networking Concepts, Issues and solutions, Mahbub Hassan, Raj Jain, First Edition, PHI, 2005.

CSE510	**Parallel Algorithms**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Algorithmic Analysis, Advanced Computer Architecture				
Objectives:	To provide fundamentals in design, analysis, and implementation, of high performance computational science and engineering applications that serve the foundations for the advanced computer architectures, parallel algorithms, parallel languages, and performance-oriented computing.				
Expected Outcome:	Students will develop knowledge and skills concerning: 1. The key factors affecting performance of CSE applications 2. mapping of applications to high-performance computing systems, and 3. Hardware/software co-design for achieving performance on real-world applications.				
Module I	Parallel Algorithm Design				
Boundary Value Problem, Finding the Maximum, Complexity measure for parallel algorithms.					
Module II	Parallel Combinatorial Algorithms				
Permutations with and without repetitions, combinations, derangements.					
Module III	Parallel Searching Algorithms				
Maximum/ minimum, median, k^{th} largest/smallest element, Parallel sorting algorithms.					
Module IV	Parallel Graph Algorithms				
Parallel graph search and tree traversal algorithms, parallel algorithms for connectivity problems, parallel algorithms for path problems.					
Module V	Programming for Parallel Algorithms				
Shared-Memory Programming with OpenMP, Message-Passing Programming, Performance Analysis					
Reference Books					
1. Ananth Grama, Anshul Gupta, George Karypis, and, Vipin Kumar, Introduction to Parallel Computing, 2nd edition, Addison-Welsey, 2003. 2. David A. Bader (Ed.), Petascale Computing: Algorithms and Applications, Chapman & Hall/CRC Computational Science Series, 2007.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

CSE517	**Distributed Systems**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Computer Architecture, Advanced Operating Systems				
Objectives:	To provide the fundamentals for the distributed systems that serve foundation for the advanced studies in the area of distributed systems.				
Expected Outcome:	On completion of this course the student would be able to deal distributed databases, file systems and also know about the languages for distributed systems.				
Module I	Introduction				
Fundamental issues in distributed systems, Distributed System Models and Architectures, Classification of Failures in Distributed Systems, Basic Techniques for Handling Faults in Distributed Systems.					
Module II	Time and Global States				
Logical clocks and physical clocks, events, process states, global states; Inter process Communication- Distributed Mutual Exclusion, Leader Election, Distributed Deadlock Detection, Remote Procedure Calls, Broadcast Protocols.					
Module III	Naming in Distributed Systems				
Name services and the DNS- Directory Services-X 500 protocol; Distributed File System and implementation; coordination and agreement.					
Module IV	Transactions and Concurrency Control				
Distributed transaction-concurrency control-transaction recovery; replication-transaction with replication; Distributed Shared Memory.					
Module V	Mobile and Ubiquitous Computing				
Context aware computing; web services; distributed coordination of services; case study on CORBA					
Reference Books					
1. Randy Chow, and Theodore Johnson. Distributed Operating Systems and Algorithms. Addison- Wesley, 1997. 2. G. Coulouris, J. Dollimore, and and T. Kindberg, "Distributed Systems: Concepts and Designs, Fourth Edition, Addison Wesley, 2005. 3. Mukesh Singhal, and N. G. Shivaratri. Advanced Concepts in Operating Systems, Distributed, Database, and Multiprocessor Operating Systems, McGraw Hill, 1994. 4. Vijay K. Garg, Elements of Distributed Computing, Wiley & Sons, 2002. 5. Relevant papers from various IEEE and ACM Transactions/Journals and Conference Proceedings.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

Program Elective-II

CSE512	**Advances in Complier Design**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Computer Architecture, Advanced Algorithmic Analysis				
Objectives:	To improve programming skills by learning how a compiler works and also provides knowledge in design and implement the parts of a compiler for a sample programming language.				
Expected Outcome:	At the end of the course students should able to 1. Analysis code optimization techniques 2. Design, code, test, and debug efficiency of simple programs 3. Implement a phases of compiler for a sample language				
Module I	Introduction				
Processor architectures, code generation, intermediate representations, tools, Foundations, e.g., control and data dependence, data flow analysis					
Module II	Optimizations				
Overview of optimizations, SSA and its construction, SSA based optimizations, Memory SSA, Firm, Chi functions in lazy memory SSA based analysis, Firm, Compiler generators.					
Module III	Semitics				
Generating semantics modules from Natural Semantics using RML, Automatically generating code generators.					
Module IV	Code Generation				
Register allocation. Mutation scheduling. Loop scheduling / Software pipelining. Memory hierarchy optimization.					
Module V	Languages				
Description languages for irregular architectures, DSPs. Special code generation problems for irregular architectures, DSPs.					
Reference Books					
1. Steven Muchnick: Advanced Compiler Design and Implementation. Morgan Kaufmann, 1997. 2. Alfred Aho, Ravi Sethi, Jeffrey Ullman, Monica Lam: 21st Century Compilers. Addison-Wesley, 2004. 3. Keith Cooper, Linda Torczon: Engineering a Compiler. Morgan Kaufmann, ,2003 4. Y.N. Srikant, P. Shankar (ed.): The compiler design handbook: optimizations and machine code generation, CRC Press, 2003.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

CSE513	**Data Compression**	L 2	T 0	P 0	C 2
Version No.	1.0				
Prerequisite	Higher Mathematics				
Objectives:	This subject aims to cover basic concepts of Graph theory				
Expected Outcome:	The students would be able to understand and explain fundamentals of Graph Theory their applications.				
Module I	Introduction				
Compression techniques, lossless compression, lossy compression, measures of performance, modeling and coding. **Mathematical preliminaries** Overview, introduction to information theory, models, physical models, probability models, markov models.					
Module II	Huffman coding andArithmetic codes				
Good codes, Huffman coding algorithm, minimum variance Huffman codes, length of Huffman codes, extended Huffman codes, non binary Huffman codes, adaptive Huffman codes, applications. **Arithmetic codes** Overview, coding a sequence, generating a binary code, compression of Huffman and arithmetic coding, applications.					
Module III	Lossless image compression				
Introduction, facsimile encoding, run length encoding, progressive image transmission, other approaches.					
Module IV	Vector quantization				
Introduction, advantages lbg-algorithm, empty cell problem, tree structured vector quantizer, other vector quantization schemes.					
Module V	Differential coding				
Overview, introduction, basic algorithm dpcm, adpcm, delta modulation, cfdf, speech coding. Transform coding:Different transforms, quantization and coding of transforms, application to image compression. Wave let transforms and data compression introduction, transform coding, dtwt for image compression, audio compression, and video coding using multi-resolution techniques					
Reference Books					
1. Khalid Sayood : Introduction To Data Compression: Second Edition Jan 1996, Morgan Kaufmann Publications. (Chapters 1.1 To 1.2, 2.1 To 2.3, 3.1 To 3.6, 4.1 To 4.6, 6.1 To 6.5, 9.1 To 9.6, 10.1 To 10.7, 12.1 To 12.6) 2. Ralf Steinmetz and Klara Nahrsedt, Multimedia Computing and Communication and Applications, Prentice Hall Intl. 1995. 3. Raghuveer M. Rao, Wavelet Transforms: Introduction to Theory and Applications, Addison Wesley Pub. Co. Ltd. 1998.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

Course Code: CSE522	Mobile Computing	L	T	P	C
Version No. 1.0	Date of Approval: 19/06/15	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Course Objectives

1. Understand system requirements for mobile applications.
2. Generate suitable design using specific mobile development frameworks.
3. Generate mobile application design.
4. Implement the design using specific mobile development frameworks.
5. Deploy the mobile applications in marketplace for distribution.

Course Outcomes

At the end of the course, students will be able to:

1. Describe the requirements for mobile applications.
2. Explain the challenges in mobile application design and development.
3. Develop design for mobile applications for specific requirements.
4. Implement the design using Android SDK.
5. Implement the design using Objective C and iOS.
6. Deploy mobile applications in Android and iPhone marketplace for distribution.

Course Content

Unit I: INTRODUCTION **8 Hours**

Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications.

Unit II: BASIC DESIGN **8 Hours**

Introduction – Basics of embedded systems design – Embedded OS - Design constraints for mobile applications, both hardware and software related – Architecting mobile applications – User interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

Unit III: **ADVANCED DESIGN** **8 Hours**

Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications.

Unit IV: ANDROID **8 Hours**

Introduction – Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration with social media applications.

Unit V: IOS **8 Hours**

Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wifi - iPhone marketplace.

Text Books

1. Charlie Collins, Michael Galpin and Matthias Kappler, ―Android in Practice‖, DreamTech, 2012.
2. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, ―Beginning iOS 6 Development: Exploring the iOS SDK‖, Apress, 2013.
3. http://developer.android.com/develop/index.html.

Reference Books

1. James Dovey and Ash Furrow, ―Beginning Objective C‖, Apress, 2012.
2. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012.
3. Reto Meier, ―Professional android Development‖, Wiley-India Edition, 2012

CSE515	**Natural Language Processing**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Higher Mathematics, Advanced Database Systems				
Objectives:	To introduce the fundamental techniques of natural language processing, to develop an understanding of the limits of those techniques and of current research issues, and to evaluate some current and potential applications.				
Expected Outcome:	At the end of the course students should 1. be able to discuss the current and likely future performance of several NLP applications, such as machine translation and email response 2. Be able to describe briefly a fundamental technique for processing language for several subtasks, such as morphological analysis, parsing, word sense disambiguation etc. 3. understand how these techniques draw on and relate to other areas of (theoretical) computer science, such as formal language theory, formal semantics of programming languages, or theorem proving.				
Module I	Introduction				
Introduction to the Morphology, Syntax, Semantics by linking the "linguistics view" (computational linguistics) with the "artificial intelligence view" (natural language processing).					
Module II	Morphology				
Analysis and generation of language on word level: e.g. problems with compounding and idiomatic phrases, homophonous strings as well as loan words and their processing using e.g. finite state automata as well as semantic networks. Ambiguities in words like "pen" and "pipe", but will also discuss some complex strings.					
Module III	Syntax				
Analysis and generation of language on phrasal and sentence level: e.g. applications such as machine translation and grammar checking and the processing using phase structure grammars as well as unification based formalisms and relating those formalisms to recursive transition networks (RTNs) as well as augmented transition networks (ATNs).					
Module IV	Semantics				
Language ambiguities on the level of "meaning": represented by case structures and conceptual dependency structures. We will look at famous utterances such as: Colourless green ideas sleep furiously. And will discuss why the machine runs into problems during analysis, and how these problems can be overcome.					
Module V	Applications of NLP				
Machine Translation, Grammar Checkers Dictation, Automatic Document Generation, NL Interfaces					
Reference Books					
1. Daniel Jurafsky, James H. Martin "Speech and Language Processing" Prentice Hall, 2001 2. Chris Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing"*, MIT Press. Cambridge, MA: May 1999.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

Program Elective-III

<table>
<tr><td>CSE518</td><td>Interaction Design</td><td>L
3</td><td>T
0</td><td>P
0</td><td>C
3</td></tr>
<tr><td>Version No.</td><td colspan="5">1.0</td></tr>
<tr><td>Prerequisite</td><td colspan="5">NIL</td></tr>
<tr><td>Objectives:</td><td colspan="5">To provide the fundamentals for interaction design which is important aspect of designing any product, involving interaction.</td></tr>
<tr><td>Expected Outcome:</td><td colspan="5">On completion of this course the student would be able to analyse and design an efficient interaction model.</td></tr>
<tr><td>Module I</td><td colspan="4">Introduction</td><td></td></tr>
<tr><td colspan="6">Introduction, Good and poor design, What is interaction design? , The user experience
The process of interaction design, Interaction design and the user experience.
Understanding the problem space , Conceptualizing the design space , Theories, models and frameworks</td></tr>
<tr><td>Module II</td><td colspan="4">Understanding users,Understanding the problem space</td><td></td></tr>
<tr><td colspan="6">What is cognition? Applying knowledge from the physical world to the digital world , Conceptual frameworks for cognitionConceptualizing the design space ,Theories, models and frameworks.</td></tr>
<tr><td>Module III</td><td colspan="4">Affective aspects,Interfaces and interactions</td><td></td></tr>
<tr><td colspan="6">What are affective aspects? Expressive interfaces and positive emotions , Frustrating interfaces and negative emotions , Persuasive technologies , Anthropomorphism ,Interface agents, virtual pets and interactive toys.</td></tr>
<tr><td>Module IV</td><td colspan="4">Data gathering, Data analysis, interpretation, and presentation</td><td></td></tr>
<tr><td colspan="6">Four key issues, Data recording, Interviews, Questionnaires, Observation, choosing and combining techniques,Qualitative and quantitative, Simple quantitative analysis ,Simple qualitative analysis ,Using Theoretical Frameworks ,Tools to support analysis ,Presenting your findings</td></tr>
<tr><td>Module V</td><td colspan="4">Identifying needs,Design, prototyping and construction</td><td></td></tr>
<tr><td colspan="6">What, how, and why? , What are requirements? , Data gathering for requirements, Data analysis, Task description, Task analysis, Prototyping and construction.</td></tr>
<tr><td colspan="2">Reference Books</td><td colspan="4"></td></tr>
<tr><td colspan="6">“Interaction Design, Beyond Human Computer Interaction”,JennyPreece , Yvonne Rogers , & Helen Sharp ,John Wiley
Human-Computer Interaction (3rd Edition) Alan Dix , Janet E.Finlay , Gregory D. Abowd , Russell Beale. Pearson Education
Designing the User Interface: Strategies for Effective Human-Computer Interaction ,Ben Shneiderman , Catherine Plaisant , Maxine Cohen , Steven Jacobs,Pearson Education.</td></tr>
<tr><td colspan="2">Mode of Evaluation</td><td colspan="4">Quiz/Assignment/ Seminar/Written Examination</td></tr>
<tr><td colspan="6">Recommended by the Board of Studies on:</td></tr>
<tr><td colspan="6">Date of Approval by the Academic Council:</td></tr>
</table>

CSE554	**Data Compression Lab**
Version No.	1.0
Prerequisite	Higher Mathematics
Objectives:	To provide hands on experience on compression techniques
Expected Outcome:	After completion of this course the students will be able develop the project to compress any kind of files.

Guidelines for experiments

Use any language of your choice.

1. Write a program which inputs a string of 1s and 0s and compresses the 0s using the run-length compression technique.
2. Write a program to find the format of file? Is this a compressed format? What is the size of the file in bytes?
3. Develop a program to implement Arithmetic coding.
4. Develop a program to compress file using Huffman coding.
5. Implement the run-length encoding to compress file data and uncompress.
6. Write program to find difference between compression file and non compress file size.
7. Develop a program using vector quantization technique.
8. Write a program to compress and uncompress file using adaptive Huffman coding.
9. Develop a program to compress image using Lossy DPCM Algorithmand evaluate performance of DPCM Algorithm.
10. Write a program to implement Huffman data compression algorithm to generate Prefix codes and encoded text.
    a. Count of character frequencies.

    b. Construction of prefix code.

    c. Encoding the text.

11. Develop a program to implement dtwt compression techniques.
12. Develop a program for compress the video file using the video compression technique.
13. Develop a program to implement Wave let transform technique.
14. Develop program to implement transform coding.

Reference Book	
Manual Prepared by faculty	
Mode of Evaluation	Lab examination / viva voce examination
Recommended by the Board of Studies on:	
Date of Approval by the Academic Council:	

Course Code: Elective-3	Real Time Systems	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Elective-3 : Real Time Systems

Course Objectives:

1. Real-time scheduling and schedulability analysis
2. Formal specification and verification of timing constraints and properties
3. Design methods for real-time systems
4. Development and implementation of new techniques to advance the state-of-the-art real-time systems research

Course Outcomes:

1. Apply principles of real time system design techniques to develop real time applications.
2. Make use of database in real time applications.
3. Make use of architectures and behavior of real time operating systems.
4. Apply evaluation techniques in application.

Course Content

Module 1: Real Time System and Scheduling

Introduction– Structure of a Real Time System –Task classes – Performance Measures for Real Time Systems – Estimating Program Run Times – Issues in Real Time Computing – Task Assignment and Scheduling – Classical uniprocessor scheduling algorithms –Fault Tolerant Scheduling.

Module 2: Software Requirements Engineering

Requirements engineering process – types of requirements – requirements specification for real time systems – Formal methods in software specification – structured Analysis and Design – object oriented analysis and design and unified modelling language – organizing the requirements document – organizing and writing documents – requirements validation and revision.

Module 3: Intertask Communication and Memory Management

Buffering data – Time relative Buffering- Ring Buffers – Mailboxes – Queues – Critical regions – Semaphores – other Synchronization mechanisms – deadlock – priority inversion – process stack management – run time ring buffer – maximum stack size – multiple stack arrangement – memory management in task control block - swapping – overlays – Block page

management – replacement algorithms – memory locking – working sets – real time garbage collection – contiguous file systems.

Module 4: Real Time Databases

Real time Databases – Basic Definition, Real time Vs General Purpose Databases, Main Memory Databases, Transaction priorities, Transaction Aborts, Concurrency control issues, Disk Scheduling Algorithms, Two– phase Approach to improve Predictability – Maintaining Serialization Consistency – Databases for Hard Real Time Systems.

Module 5: Evaluation Techniques and Clock Synchronization

Reliability Evaluation Techniques – Obtaining parameter values, Reliability models for Hardware Redundancy–Software error models. Clock Synchronization–Clock, A Nonfault–Tolerant Synchronization Algorithm – Impact of faults – Fault Tolerant Synchronization in Hardware – Fault Tolerant Synchronization in software.

Text books:

1. Real-Time Systems: Theory and Practice, 1e by MALL
2. Real-Time Systems, 1eby LIU

Reference Books:

1. Real Time Systems by Liu Jane W.S. Pearson LPE
2. REAL TIME SYSTEMSby C.M. Krishna and K.G. Shin

CSE555	**Web Services Lab**	0	0	2	1
Version No.	1.0				
Prerequisite	Operating System and Networking Lab				
Objectives:	To provide hands on experience on the basics of .NET, C#.				
Expected Outcome:	After completion of this course the students will be able to undergo projects in .NET				

Guidelines for experiments

1. Tutorial: Basics of Web Service & C#, ASP.NET Fundamentals.
2. Development of a Hello World web service in C# on Microsoft Visual Studio.
3. Development of a .NET web client application to consume the .NET web service.
4. Development of a Java Client Application for consuming the Java web service.
5. Development of a Java Client Application for consuming the .NET web service.
6. Create a Web Service in to demonstrate the following:

a. Linear Search, Binary Search

b. Selection Sort, Insertion Sort

7. Develop a Web Service without using Visual Studio.NET IDE and Demonstrate how to consume the web service. (Using SOAP, UDDI, WSDL)
8. Create an Employee Database in MS-Access (eno, ename, DOB, designation, address, phone, mail). Develop a Web Service in .NET to return the specific employee details which is taken from Web Client.
9. Develop a Web Service in .NET to perform the encryption and decryption for files using DES algorithm.
10. Develop a Web Service in for currency conversion.
11. Develop a Web Service in .NET for file compression & decompression using J# Utilities.
12. Demonstrate Remoting Application using .NET. (using TCP & HTTP channels)
13. Create a Web Service to calculate CGPA for at least 3 semesters. (4 subjects + 2 labs for each semester) Calculate GPA for a semester and then CGPA for all the semester.
14. Develop a Web Site which demonstrates ASP.NET authentication, authorization and different validation controls.(also include link to different web services).

Reference Book	
Manual Prepared by faculty	
Mode of Evaluation	Lab examination / viva voce examination
Recommended by the Board of Studies on:	
Date of Approval by the Academic Council:	

Program Elective-IV

CSE511	**Big Data Analytics**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Operating Systems, Advanced Computer Networks				
Objectives:	To understand the competitive advantages of big data analytics • To understand the big data frameworks • To learn data analysis methods • To learn stream computing				
Expected Outcome:	• Understand how to leverage the insights from big data analytics • Analyze data by utilizing various statistical and data mining approaches • Perform analytics on real-time streaming data • Understand the various NoSql alternative database models				
Module I	INTRODUCTION TO BIG DATA				
Big Data – Definition, Characteristic Features – Big Data Applications - Big Data vs Traditional Data - Risks of Big Data - Structure of Big Data - Challenges of Conventional Systems - Web Data – Evolution of Analytic Scalability - Evolution of Analytic Processes, Tools and methods - Analysis vs Reporting - Modern Data Analytic Tools					
Module II	HADOOP FRAMEWORK				
Distributed File Systems - Large-Scale FileSystem Organization – HDFS concepts - MapReduce Execution, Algorithms using MapReduce, Matrix-Vector Multiplication – Hadoop YARN					
Module III	DATA ANALYSIS				
Statistical Methods: Regression modelling, Multivariate Analysis - Classification: SVM & Kernel Methods - Rule Mining - Cluster Analysis, Types of Data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Clustering Methods, Clustering High Dimensional Data - Predictive Analytics – Data analysis using R.					
Module IV	MINING DATA STREAMS				
Streams: Concepts – Stream Data Model and Architecture - Sampling data in a stream - Mining Data Streams and Mining Time-series data - Real Time Analytics Platform (RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.					
Module V	BIG DATA FRAMEWORKS				
Introduction to NoSQL – Aggregate Data Models – Hbase: Data Model and Implementations – Hbase Clients – Examples – .Cassandra: Data Model – Examples – Cassandra Clients – Hadoop Integration. Pig – Grunt – Pig Data Model – Pig Latin – developing and testing Pig Latin scripts. Hive – Data Types and File Formats – HiveQL Data Definition – HiveQL Data Manipulation – HiveQL Queries					
Reference Books					

1. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics”, Wiley and SAS Business Series, 2012.
2. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", 2013.
3. Learning R – A Step-by-step Function Guide to Data Analysis, Richard Cotton, O‟Reilly Media, 2013.
4. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer, Second Edition, 2007.
5. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley,

<table>
<tr><td colspan="2">2013.
6. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.</td></tr>
<tr><td>Mode of Evaluation</td><td>Quiz/Assignment/ Seminar/Written Examination</td></tr>
<tr><td colspan="2">Recommended by the Board of Studies on:</td></tr>
<tr><td colspan="2">Date of Approval by the Academic Council:</td></tr>
</table>

Elective 4	**Transaction Procesing**	L	T	P	C
Version No. 1.0	**Date of Approval: 19/06/15**	3	0	0	3
Prerequisite/Exposure					
Co-requisites					

Course Objectives:

1. To cover transaction processing systems which includes serializability theory,
2. Students understand the concurrency control, recovery protocols
3. Students able to develop the distributed commit protocols, replication, parallelism.
4. Students able to develop the distribution design, and federated and multi-databases.

Course Outcomes:

1. Students able to deal issues related to transaction processing and able to implement a state of art transaction processing subsystem.
2. Students can use current techniques, skills, and tools necessary for computing practice.
3. An ability to use and apply current technical concepts and practices in the core information technologies.
4. An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.

Course Content

ModuleI: Introduction

Intro to transactions, Problems that transactions solve, OLTP, e-Commerce, TP Monitors; Database Structure -- File Organization and Access Paths: Reference architectures, Memory hierarchies, File organization, Access paths and indexes.

Module II: Serializability and Conflict Equivalence

Transaction semantics, Partial and full orderings, Conflict serializability, Commutativity; Spatial Indexing Scientific Databases: Space filling curves, Quad-trees, HTM; Transaction Models: TxN models, Write-ahead logging, Undo, Redo, Undo/Redo protocols, Steal and Force; Introduction to ARIES:LSNs, ARIES Principles, Page-oriented logging, Checkpoints.

Module III: Locking Schedulers

Lock compatibility, Wormhole transactions, Two-Phase Locking (2PL); Slowdowns and Deadlock: Convoys, Priority Inversion, Deadlock detection and avoidance; Phantoms and Locking Exotics: Granular locking, Escrow locking, Optimistic locking, Timestamp ordering.

Module IV: Distributed Commit Protocols

Two-phase commit, presumed abort/commit optimizations, Three-phase commit; Voting Protocols and Replication: Quorums, Dynamic Voting, Optimizations; Replication -- Advanced Topics: Anti-entropy protocols, Epidemic replication, Weak consistency replication, Distribution design.

Module V: Isolation Levels in SQL

Isolation Degrees, Manual tuning of Isolation/Performance tradeoffs; Image/Object Representations and Query Optimization: Tesselations, Representing Spatial Objects, Query Evaluation Plans, Histograms; Joins, Views and Network Joins: Isolation Degrees, Manual tuning of Isolation/Performance tradeoffs

Text Books

1. Weikum, Gerhard and Gottfried Vossen. Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control. Morgan Kaufmann Publishers. Copyright 2002
2. Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. Database Systems Concepts, 4/e.,McGraw-Hill Publishers.Copyright 2001

Reference Books

1. Garcia-Molina, H., J. D. Ullman and J. Widom. Database System Implementation, Prentice-Hall, Inc., 2000.

CSE516	**Web Services**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Computer Networks, Advanced Database Systems				
Objectives:	To provide fundamentals on SOA, SOAP UDDI and XML that lays foundations for the advanced studies in the area of web services.				
Expected Outcome:	After completion of this course the students able to perform project in the area of XML				
Module I	SOA: (Service Oriented Architecture)				
Introduction to Services - Bind, Publish, Find – Framework for SOA – Web Services (A Realization of SOA) - Web Services Architecture (Transport Services, Messaging Services, Service Description, Discovery Services, Quality of Service), Interoperability – REST (Representational State Transfer) Services.					
Module II	XML Basics				
XML Messaging, SOAP, UDDI and WSDL – Basics of XML – XML-RPC Essentials – Real life web services – Standards of Web Service Stack – Web Services Vendor Landscape, Building & Consuming XML Web Services in .NET, State Management.					
Module III	SOAP: Simple Object Access Protocol				
Introduction to SOAP & XML – SOAP Specification – messages, Data Encoding, Data types – Writing SOAP Web Services – Discovering SOAP Services.					
Module IV	UDDI: Universal Description, Discovery and Integration				
Overview – UDDI Business Registry (UBR) – UDDI Model (UDDI Data Structures, Keys, APIs, Nodes and Registries) - UDDI Implementations.					
Module V	WSDL: Web Service Description Language				
WSDL Specification – Basic WSDL Example - Operations, Bindings, Service – Invocation Tools – XML Schema Data Typing, Case Studies					
Reference Books					
1. Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More by SanjivaWeerawarana, Francisco Curbera, Frank Leymann, Tony Storey, Donals F. Ferguson , Prentice Hall PRT, 2005. 2. XML Web Services for ASP.NET by Bill Evjen, Wiley Publishing Inc, 2002. 3. Web Services Essentials Distributed Applications with XML-RPC, SOAP, UDDI & WSDL by Ethan Cerami, O'Reilly , First Edition, February 2002. 4. Programming Web Services with SOAP by James Snell, O'Reilly First Edition Dec 2001. 5. Web Services Theory & Practice by Anura Guruge, Digital Press, 2004. 6. Executive's Guide to Web Services by Eric A. Marks & Mark. J. Werrell, John Wiley & Sons, 2003.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					

CSE511	**Grid and Cluster Computing**	L 3	T 0	P 0	C 3
Version No.	1.0				
Prerequisite	Advanced Operating Systems, Advanced Computer Networks				
Objectives:	To create a framework that effectively makes use of the computational powers and resources of the computer systems within that framework.				
Expected Outcome:	On completion of this course the students will be able to know about various technologies and tools used to create a grid and can create a framework that can effectively utilize the resources and computational powers.				
Module I	Introduction				
The Grid – Beginning of the Grid – Evolution of the Grid – A CommModuley Grid Model – Building Blocks of Grid – An overview of Grid Business Areas - Grid Application – Grid infrastructure.					
Module II	Grid Computing Organizations and their Roles				
Organizations developing Grid standards and the Global Grid Forum – Organizations developing Grid Computing Toolkits and Frameworks – Grid Computing Anatomy – Grid Computing Road Map.					
Module III	New Generation of Grid Computing Applications				
Service Oriented Architecture – Web Service Architecture – XML, Related Technologies – XML Messages and Enveloping – Service Message Description Mechanisms – Relationship between Web Services and Grid Services.					
Module IV	Grid Computing Technology				
Open Grid Services Architecture (OGSA) – OGSA Platform Components – Open Grid Services Infrastructure (OGSI) - OGSA Basic Services.					
Module V	Grid Computing Toolkits				
GLOBUS GT3 Toolkit: Architecture, Programming Model, implementation, High Level Services					
Reference Books					
7. Joshy Joseph, Craig Fellenstein, Grid Computing, Pearson Education, 2004. 8. Fran Berman, Geoferry C. Fox, Antthony J.G. Hey, Grid Computing Making the Global Infrastructure a Reality, Wiley Series in Communications Networking & Distributed Systems, 2000.					
Mode of Evaluation	Quiz/Assignment/ Seminar/Written Examination				
Recommended by the Board of Studies on:					
Date of Approval by the Academic Council:					