HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

(Approved by AICTE, New Delhi, Accredited by NAAC with 'A' Grade)

COIMBATORE 641 032

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.E CSE -CURRICULUM AND SYLLABI

REGULATIONS 2016

Common to all M.E / MCA/ MBA DEGREE PROGRAMMES

(CHOICE BASED CREDIT SYSTEM)

INDEX

REGULATION – 2016

M.E. COMPUTER SCIENCE AND ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi and Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE 641 032

REGULATION- 2016

M. E COMPUTER SCIENCE AND ENGINEERING

VISION

To provide an excellence for individuals to develop technologically superior socially conscious and nationally responsible citizens

MISSION

- To develop competent Computer Science and Engineering professionals with knowledge in current technology.
- To mould them to attain excellent leadership qualities there by making them excel in their careers.
- To inspire and nurture students to come out with innovation and creativity solutions meeting the societal needs.

SALIENT FEATURES OF THE SYLLABUS

1. **"Advanced Internet of Things"** subject is added in the III semester.

   The Internet of Things (IoT) is one of the hottest topics in the technology sector, and with good reason. It influences the interaction of technological, economic, social, societal, and individual changes, and analysts and market researchers estimate that by the year 2020.
2. **Open Elective** subjects are included in the III semester, so that students can choose any interested subjects from other departments, which improve their inter-disciplinary subject knowledge.
3. **Elective papers** are very specific, specialized, advanced and supportive to the discipline/ subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the

engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: To acquire knowledge in the latest technologies and innovations and an ability to identify, analyze and solve problems in computer engineering.

PEO2: To be capable of modeling, designing, implementing and verifying a computing system to meet specified requirements for the benefit of society.

PEO3: To possess critical thinking, communication skills, teamwork, leadership skills and ethical behavior necessary to function productively and professionally.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to apply, design and develop principles of software engineering, networking and data base concepts for computer- based systems in solving engineering problems.

PSO2: An ability to understand, design and code engineering problems using programming skills.

CURRICULUM AND SYLLABI

SEMESTER I

S.No.	Course Code	Course Title	L	T	P	C	CIA	ESE	TOTAL
THEORY									
1	16MA1122	Advanced Mathematics For Computing	3	1	0	4	40	60	100
2	16CP1201	Design and Management of Computer Networks	3	0	0	3	40	60	100
3	16CP1202	Advanced Data Structures and Algorithms	3	1	0	4	40	60	100
4	16CP1203	Multicore Architectures	3	0	0	3	40	60	100
5	16CP13XX	Professional Elective I	3	0	0	3	40	60	100
6	16CP13XX	Professional Elective II	3	0	0	3	40	60	100
PRACTICAL									
7	16CP1001	Advanced Data Structures Laboratory	0	0	4	2	50	50	100
8	16CP1701	Case Study - Network Design (Team Work)	0	0	4	2	50	50	100
		Total :	**18**	**2**	**8**	**24**			**800**

SEMESTER II

S.No.	Course Code	Course Title	L	T	P	C	CIA	ESE	TOTAL
THEORY									
1	16CP2201	Research Methodology	3	1	0	4	40	60	100
2	16CP2202	Advanced Databases	3	0	0	3	40	60	100
3	16CP2203	Principles of Programming Languages	3	0	0	3	40	60	100
4	16CP2204	Advanced Operating Systems	3	0	0	3	40	60	100
5	16CP23XX	**Professional Elective III**	3	0	0	3	40	60	100
6	16CP23XX	**Professional Elective IV**	3	0	0	3	40	60	100
PRACTICAL									
7	16CP2001	Advanced Databases Laboratory	0	0	4	2	50	50	100
8	16CP2702	Case Study - Operating Systems Design (Team Work)	0	0	4	2	50	50	100
		Total :	**18**	**1**	**8**	**23**			**800**

SEMESTER III

S.No.	Course Code	Course Title	L	T	P	C	CIA	ESE	TOTAL
THEORY									
1	16CP3201	Software Process and Project Management	3	1	0	4	40	60	100
2	16CP3202	Advanced Internet of Things	3	0	0	3	40	60	100
3	16CP33XX	**Professional Elective V**	3	0	0	3	40	60	100
4	16CP33XX	**Professional Elective VI (OR)**	3	0	0	3	40	60	100
	16XX34XX	**Open Elective (Optional)**							
PRACTICAL									
5	16CP3901	Project Phase - I	0	0	12	6	50	50	100
		Total :	**12**	**1**	**12**	**19**			**500**

SEMESTER IV

S.No.	Course Code	Course Title	L	T	P	C	CIA	ESE	TOTAL
PRACTICAL									
1	16CP3902	Project Phase - II	0	0	24	12	100	100	200
		Total :	**0**	**0**	**24**	**12**			**200**

LIST OF PROFESSIONAL ELECTIVES

S.No.	Course Code	Course Title	L	T	P	C	CIA	ESE	TOTAL
PROFESSIONAL ELECTIVE I									
1	16CP1301	Data Analysis and Business Intelligence	3	0	0	3	40	60	100
2	16CP1302	Formal models of software systems	3	0	0	3	40	60	100
3	16CP1303	Performance Evaluation of Computer Systems	3	0	0	3	40	60	100
4	16CP1304	Probabilistic Reasoning Systems	3	0	0	3	40	60	100
5	16CP1305	Image Processing and Analysis	3	0	0	3	40	60	100
PROFESSIONAL ELECTIVE II									
1	16CP1306	Randomized Algorithms	3	0	0	3	40	60	100
2	16CP1307	Mobile and Pervasive Computing	3	0	0	3	40	60	100
3	16CP1308	Parallel Programming Paradigms	3	0	0	3	40	60	100
4	16CP1309	Software Requirements Engineering	3	0	0	3	40	60	100
5	16CP1310	Speech Processing and Synthesis	3	0	0	3	40	60	100
PROFESSIONAL ELECTIVE III									
1	16CP2301	Machine Learning Techniques	3	0	0	3	40	60	100
2	16CP2302	Real Time Systems	3	0	0	3	40	60	100
3	16CP2303	Computer Vision	3	0	0	3	40	60	100
4	16CP2304	Network and Information Security	3	0	0	3	40	60	100
5	16CP205	Software Architectures	3	0	0	3	40	60	100

PROFESSIONAL ELECTIVE IV									
1	16CP2306	Bio-inspired Computing	3	0	0	3	40	60	100
2	16CP2307	Cloud Computing	3	0	0	3	40	60	100
3	16CP2308	Data Visualization Techniques	3	0	0	3	40	60	100
4	16CP2309	Protocols and Architecture for Wireless SensorNetworks	3	0	0	3	40	60	100
5	16CP2310	Language Technologies	3	0	0	3	40	60	100

PROFESSIONAL ELECTIVE V									
1	16CP3301	Social Network Analysis	3	0	0	3	40	60	100
2	16CP3302	Managing Big Data	3	0	0	3	40	60	100
3	16CP3303	Model Checking and Program Verification	3	0	0	3	40	60	100
4	16CP3304	Medical Image Processing	3	0	0	3	40	60	100
5	16CP3305	Software Design	3	0	0	3	40	60	100
PROFESSIONAL ELECTIVE VI									
1	16CP3306	Multi Objective Optimization Techniques	3	0	0	3	40	60	100
2	16CP3307	Information Storage Management	3	0	0	3	40	60	100
3	16CP3308	Software Quality Assurance	3	0	0	3	40	60	100
4	16CP3309	Green Computing	3	0	0	3	40	60	100
5	16CP3310	Reconfigurable Computing	3	0	0	3	40	60	100

OPEN ELECTIVE									
S.No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**C**	**CIA**	**ESE**	**TOTAL**
1	16CPX401	Mobile Application Development	3	0	0	3	40	60	100
2	16CPX402	Data Mining Techniques	3	0	0	3	40	60	100

SEMESTER	I	II	III	IV	**TOTAL**
CREDITS	**24**	**23**	**19**	**12**	**78**

Signature and Name of Chairman, BOS

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16MA1122	ADVANCED MATHEMATICS FOR COMPUTING	3	1	0	4

Course Objective

1. To apply testing of hypothesis to infer outcome of experiments.
2. To understand, develop and solve mathematical model of linear programming problems, transport and assignment problems.
3. To understand network modeling for planning and scheduling the project activities.
4. To appreciate the use of simulation techniques.
5. To introduce Number theory based computing and network security studies in Computer Science.

Unit	Description	Instructional Hours
I	**TESTING OF HYPOTHESES** Sampling distributions - Type I and Type II errors - Tests based on Normal, t, Chi-Square and F distributions for testing of mean, variance and proportions – Tests for Independence of attributes and Goodness of fit.	9+3
II	**LINEAR PROGRAMMING** Formulation – Graphical solution – Simplex method –Artificial variable Techniques - Transportation and Assignment Models.	9+3
III	**SCHEDULING BY PERT AND CPM** Network Construction – Critical Path Method – Project Evaluation and Review Technique – Resource Analysis in Network Scheduling.	9+3
IV	**SIMULATION** Discrete Event Simulation –Monte –Carlo Simulation –Stochastic Simulation –Applications to Queuing systems.	9+3
V	**NUMBER THEORY** Combinatorics-Generating functions-Applications of number theory, Mathematical logic-theory of interference, Rough set theory-soft set theory.	9+3
	Total Instructional Hours	**60**

Course Outcome

CO1: Acquire the basic concepts of Probability and Statistical techniques for solving mathematical problems which will be useful in solving engineering problems.

CO2 : Apply transportation and assignment models to find optimal solution in warehousing and travelling.

CO3 : Prepare project scheduling using PERT and CPM

CO4 : Familiarize the students with special functions and solve problems associated with engineering applications.

CO5 : Apply number theory for various applications

TEXT BOOKS:

T1 - Jay L. Devore., "Probability and Statistics For Engineering and the Sciences", Thomson and Duxbury, 2002.

T2 - Richard Johnson., "Miller & Freund's Probability and Statistics for Engineer", Prentice – Hall , Seventh Edition, 2007

T3 - Taha H.A., "Operations Research : An Introduction " 8th Edition, Pearson Education, 2008.

REFERENCE BOOKS :

R1 - Gupta S.C. and Kapoor V.K., "Fundamentals of Mathematical Statistics", Sultan chand & Sons, 2001.

R2 - Prem Kumar Gupta, D.S. Hira, "Operations Research", S.Chand & Company Ltd, New Delhi, 3rd Edition , 2008.

R3 - Paneer Selvam, "Operations Research", Prentice Hall of India, 2002.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1201	DESIGN AND MANAGEMENT OF COMPUTER NETWORKS	3	0	0	3

Course Objective

1. To understand the basics of network management.
2. To analyze the various aspects of requirements.
3. To interpret the flow analysis.
4. To understand the concepts of network architecture
5. To apply the concepts of network management for network design..

Unit	Description	Instructional Hours
	INTRODUCTION TO NETWORK MANAGEMENT	
I	Overview of Analysis, Architecture and Design Process-System Methodology, Service methodology, Service Description - Service characteristics - Performance Characteristics - Network supportability - Requirement analysis – User Requirements – Application Requirements – Device Requirements – Network Requirements – Other Requirements - Requirement specification and map.	9
	REQUIREMENTS ANALYSIS	
II	Background- User requirements – Application requirements – Network requirements - Requirement Analysis Process – Gathering and Listing Requirements- Developing service metrics – Characterizing behavior – Developing RMA requirements – Developing delay Requirements - Developing capacity Requirements.	9
	FLOW ANALYSIS	
III	Flows - Identifying and developing flows – Data sources and sinks – Flow models- Flow prioritization – Flow specification algorithms – Example Applications of Flow Analysis.	9
	NETWORK ARCHITECTURE	
IV	Background - Architecture and design – Component Architectures – Reference Architecture – Architecture Models – System and Network Architecture –Addressing and Routing Fundamentals – Addressing Mechanisms – Addressing Strategies – Routing Strategies – Network Management Architecture – Network Management Mechanisms Performance Architecture – Performance Mechanisms – Security and Privacy Architecture – Planning security and privacy Mechanisms.	9
	NETWORK DESIGN	
V	Design Concepts – Design Process - Network Layout – Vendor, Equipment and Service Provider Evaluations Design Traceability –Network Layout - Bridging, Switching and Routing Protocols-Physical Network Design – Selecting Technologies and Devices for Campus and Enterprise Networks – Optimizing Network Design.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the basics of network management.
CO2: Discuss the various aspects of requirements.
CO3: Analyze the various flow.
CO4: Design and implement network architecture.
CO5: Design various types of networks.

REFERENCE BOOKS :

R1- Network Analysis, Architecture, and Design By James D. McCabe, Morgan Kaufmann, Third Edition, 2007.ISBN-13: 978-0123704801
R2. Computer Networks: A Systems Approach by Larry L. Peterson, Bruce S. Davie - 2007, Elsevier Inc.
R3- Top-down Network Design: [a Systems Analysis Approach to Enterprise Network Design] By Priscilla Oppenheimer, Cisco Press , 3rd Edition, ISBN-13: 978-1-58720- 283-4 ISBN-10: 1- 58720-283-2
R4- Integrated Management of Networked Systems: Concepts, Architectures, and Their Operational Application (The Morgan Kaufmann Series in Networking), Heinz-Gerd Hegering, Sebastian Abeck, and Bernhard Neumair, 1999.
R5- "Network Design and Management" – by Steven T.Karris, Orchard publications, Second edition, Copyright 2009, ISBN 978-1-934404-15-7
R6- "Network Design, Management and Technical Perspective", Teresa C. Mann-Rubinson and Kornel Terplan, CRC Press, 1999

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1202	ADVANCED DATA STRUCTURES AND ALGORITHMS	3	1	0	4

Course Objective

1. To understand the principles of iterative and recursive algorithms.
2. To learn the graph search algorithms.
3. To learn the dynamic programming and greedy algorithm
4. To learn the principles of shared and concurrent objects.
5. To learn concurrent data structures.

Unit	Description	Instructional Hours
I	**ITERATIVE AND RECURSIVE ALGORITHMS** Iterative Algorithms: Measures of Progress and Loop Invariants-Paradigm Shift: Sequence of Actions versus Sequence of Assertions- Steps to Develop an Iterative Algorithm-Different Types of Iterative Algorithms--Typical Errors-Recursion-Forward versus Backward- Towers of Hanoi-Checklist for Recursive Algorithms-The Stack Frame-Proving Correctness with Strong Induction-Examples of Recursive Algorithms-Sorting and Selecting Algorithms-Operations on Integers-Ackermann's Function- Recursion on Trees-Tree Traversals- Examples- Generalizing the Problem - Heap Sort and Priority Queues-Representing Expressions.	12
II	**OPTIMISATION ALGORITHMS** Optimization Problems-Graph Search Algorithms-Generic Search-Breadth-First Search-Dijkstra's Shortest-Weighted-Path -Depth-First Search-Recursive Depth-First Search-Linear Ordering of a Partial Order- Network Flows and Linear Programming-Hill Climbing-Primal Dual Hill Climbing-Steepest Ascent Hill Climbing-Linear Programming-Recursive Backtracking-Developing Recursive Backtracking Algorithm- Pruning Branches-Satisfiability	12
III	**DYNAMIC PROGRAMMING AND GREEDY ALGORITHMS** Dynamic Programming-Rod cutting- Matrix-chain multiplication- Elements of dynamic programming- Longest common sequence- Optimal binary search trees. Greedy Algorithms-An activity selection problem-Elements of greedy strategy-Huffman codes- Matroids and greedy methods- A task scheduling problem as a matroid.	12
IV	**SHARED OBJECTS AND CONCURRENT OBJECTS** Shared Objects and Synchronization -Properties of Mutual Exclusion-The Moral- The Producer–Consumer Problem -The Readers–Writers Problem-Realities of Parallelization-Parallel Programming- Principles-Thread Solutions-The Filter Lock-Fairness-Lamport's Bakery Algorithm-Bounded Timestamps-Lower Bounds on the Number of Locations-Concurrent Objects- Concurrency and Correctness-Sequential Objects-Quiescent Consistency- Sequential Consistency-Linearizability - Progress Conditions- The Java Memory Model- Register constructions	12
V	**CONCURRENT DATA STRUCTURES** Practice-Linked Lists-The Role of Locking-List-Based Sets-Concurrent Reasoning- Coarse- Grained Synchronization-Fine-Grained Synchronization-Optimistic Synchronization- Lazy Synchronization-Non-Blocking Synchronization-Concurrent Queues and the ABA Problem- Queues-A Bounded Partial Queue-An Unbounded Total Queue-An Unbounded Lock-Free Queue- Memory Reclamation and the ABA Problem- Dual Data Structures- Concurrent Stacks and Elimination- An Unbounded Lock-Free Stack- Elimination-The Elimination Backoff Stack.	12
	Total Instructional Hours	**60**

Course Outcome

CO1: Design and apply iterative and recursive algorithms.
CO2: Design and implement optimisation algorithms in specific applications.
CO3: Design and implement dynamic programming and greedy algorithm techniques.
CO4: Design appropriate shared objects and concurrent objects for applications.
CO5: Implement and apply concurrent linked lists, stacks, and queues.

REFERENCE BOOKS :

R1- Jeff Edmonds, “How to Think about Algorithms”, Cambridge University Press, 2008.
R2- M. Herlihy and N. Shavit, “The Art of Multiprocessor Programming”, Morgan Kaufmann, 2008.
R3- Steven S. Skiena, “The Algorithm Design Manual”, Springer, 2008.
R4- Peter Brass, “Advanced Data Structures”, Cambridge University Press, 2008.
R5- S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, “Algorithms” , McGrawHill, 2008.
R6- J. Kleinberg and E. Tardos, "Algorithm Design“, Pearson Education, 2006.
R7- T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms“, PHI Learning Private Limited, 2012.
R8- Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
R9- A. V. Aho, J. E. Hopcroft, and J. D. Ullman, “The Design and Analysis of Computer Algorithms”, Addison-Wesley, 1975.
R10- A. V. Aho, J. E. Hopcroft, and J. D. Ullman,”Data Structures and Algorithms”, Pearson,2006.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1203	MULTICORE ARCHITECTURES	3	0	0	3

Course Objective

1. To understand the recent trends in the field of Computer Architecture and identify performance related parameters
2. To appreciate the need for parallel processing
3. To expose the students to the problems related to multiprocessing
4. To understand the different types of multicore architectures
5. To expose the students to warehouse-scale and embedded architectures.

Unit	Description	Instructional Hours
I	**FUNDAMENTALS OF QUANTITATIVE DESIGN AND ANALYSIS** Classes of Computers – Trends in Technology, Power, Energy and Cost – Dependability – Measuring, Reporting and Summarizing Performance – Quantitative Principles of Computer Design – Classes of Parallelism - ILP, DLP, TLP and RLP - Multithreading - SMT and CMP Architectures – Limitations of Single Core Processors - The Multicore era – Case Studies of Multicore Architectures.	9
II	**DLP IN VECTOR, SIMD AND GPU ARCHITECTURES** Introduction- Vector Architecture - SIMD Instruction Set Extensions for Multimedia – Graphics Processing Units - Detecting and Enhancing Loop Level Parallelism - Case Studies.	9
III	**TLP AND MULTIPROCESSORS** Introduction- Centralized shared memory Architectures- Symmetric and Distributed Shared Memory Architectures – Cache Coherence Issues - Performance Issues – Synchronization Issues – Models of Memory Consistency – Interconnection Networks – Buses, Crossbar and Multi-stage Interconnection Networks.	9
IV	**RLP AND DLP IN WAREHOUSE-SCALE ARCHITECTURES** Introduction- Programming Models and Workloads for Warehouse-Scale Computers – Architectures for Warehouse-Scale Computing – Physical Infrastructure and Costs – Cloud Computing – Case Studies.	9
V	**ARCHITECTURES FOR EMBEDDED SYSTEMS** Features and Requirements of Embedded Systems – Signal Processing and Embedded Applications – The Digital Signal Processor – Embedded Multiprocessors - Case Studies.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the limitations of ILP and the need for multicore architectures.
CO2: Discuss the issues related to multiprocessing and suggest solutions.
CO3: Point out the salient features of different multicore architectures and how they exploit parallelism.
CO4: Critically analyze the different types of inter connection networks.
CO5: Discuss the architecture of GPUs, warehouse-scale computers and embedded processors.

REFERENCE BOOKS :

R1 - John L. Hennessey and David A. Patterson, “Computer Architecture – A Quantitative Approach”, Morgan Kaufmann / Elsevier, 5th edition, 2012.

R2 - Kai Hwang, “Advanced Computer Architecture”, Tata McGraw-Hill Education, 2003

R3 - Richard Y. Kain, “Advanced Computer Architecture a Systems Design Approach”, Prentice Hall, 2011.

R4 - David E. Culler, Jaswinder Pal Singh, “Parallel Computing Architecture : A Hardware/ Software Approach” , Morgan Kaufmann / Elsevier, 1997.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1001	ADVANCED DATA STRUCTURES LABORATORY	0	0	4	2

Course Objective

1. To learn to implement iterative and recursive algorithms.
2. To learn to design and implement algorithms using hill climbing and dynamic programming techniques.
3. To learn to implement shared and concurrent objects.
4. To learn to implement concurrent data structures.
5. To learn optimization algorithms and randomized algorithms.

Expt. No.	Description of the Experiments
1.	Implementation of graph search algorithms.
2.	Implementation and application of network flow and linear programming problems.
3.	Implementation of algorithms using the hill climbing and dynamic programming design techniques.
4.	Implementation of recursive backtracking algorithms.
5.	Implementation of randomized algorithms.
6.	Implementation of various locking and synchronization mechanisms for concurrent linked lists, concurrent queues, and concurrent stacks.
7.	Developing applications involving concurrency
8.	Implementation of graph search algorithms.
9.	Implementation and application of network flow and linear programming problems.
10.	Implementation of algorithms using the hill climbing and dynamic programming design techniques.
11.	Implementation of recursive backtracking algorithms.
12.	Realization of passive filters

Total Practical Hours 60

Course Outcome

CO1: Design and apply iterative and recursive algorithms.
CO2: Design and implement algorithms using the hill climbing and dynamic programming and recursive backtracking techniques.
CO3: Design and implement optimization algorithms and randomized algorithms for specific applications.
CO4: Design appropriate shared objects and concurrent objects for applications.
CO5: Implement and apply concurrent linked lists, stacks, and queues.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1701	CASE STUDY: NETWORK DESIGN (TEAM WORK)	0	0	4	2

Course Objective

1. To learn to implement various configurations and protocols in LAN.
2. To learn RIP and OSPF Redistribution and Dial-on-Demand Routing.
3. To learn to implement Network Security.
4. To learn Controlling Traffic Flow and Defining Access Lists.
5. To learn Configuring a fire wall and Integrating EIGRP into Existing Networks.

Expt. No.	Description of the Experiments
1.	CASE STUDY : 1 Analyzing the performance of various configurations and protocols in LAN. 1.1. Establishing a Local Area Network (LAN): The main objective is to set up a Local Area Network,concepts involved in this network are IP addressing and the Address Resolution Protocol (ARP). The required equipments are192.168.1.1 ,192.168.1.2, 192.168.1.3,Host A Host B Host C, Switch/HUB, three PC`s equipped with at least one NIC, one HUB or Switch and the necessary cables. Once the physical LAN is set up the hosts need to be configured using the ifconfig command. To verify communication among the machines the ping command is used. Next, to manipulate the routing tables at the hosts to understand how machines know where to send packets. Since the ifconfig command places a default route into the routing tables this route must be deleted. to ´blindfold˙ the machine. The ping command is used again to show that communication is no longer available. To re-establish communication the routes are put back into the routing table one host at a time. Communication is once again verified using the ping command. 1.2. Connecting two LANs using multi-router topology with static routes: The main objective is to extend routing connection by using multiple routers. The concepts include IP addressing and basic network routing principles. Connect two LANs topology. During router configuration attention is paid to the types of interfaces as additional issues are involved with setup. For example, the serial interfaces require clocking mechanisms to be set correctly. Once the interfaces are working the ping command is used to check for communication between LANs. The failure of communication illustrates the need for routes to be established inside the routing infrastructure. Static routes are used to show how packets can be transported through any reasonable route. It is run trace route on two different configurations to demonstrate the implementation of different routes. 1.3 Analyzing the performance of various configurations and protocols Original TCP versus the above modified one: To compare the performance between the operation of TCP with congestion control and the operation of TCP as implemented . The main objective is for students to examine how TCP responds to a congested network. The concepts involved in the lab include network congestion and the host responsibilites for communicating over a network. This lab requires three PC’s connected to a switch.One PC is designated as the target host and the other two PC’s will transfer a file from the targethost using FTP. A load is placed on the network to simulate congestion and the file is transfered, first by the host using the normal TCP and then by the host using the modified version. This procedure is performed multiple times to determine average statistics. The students are then asked to summarize the results and draw conclusions about the performance differences and the underlying implications for hosts operating in a network environment.

Case Study 2:

RIP and OSPF Redistribution

This case study addresses the issue of integrating Routing Information Protocol (RIP) networks with Open Shortest Path First (OSPF) networks. Most OSPF networks also use RIP to communicate with hosts or to communicate with portions of the internetwork that

2. do not use OSPF. This case study should provide examples of how to complete the following phases in redistributing information between RIP and OSPF networks, including the following topics:

• Configuring a RIP Network
• Adding OSPF to the Center of a RIP Network
• Adding OSPF Areas
• Setting Up Mutual Redistribution

Case Study 3:

Dial-on-Demand Routing

This case study should describe the use of DDR to connect a worldwide network that consists of a central site located in Mumbai and remote sites located in Chennai, Bangalore, and Hyderabad.

The following scenarios should be considered:

• Having the Central Site Dial Out

Describe the central and remote site configurations for three setups: a central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Include examples of the usage of rotary groups and access lists.

• Having the Central and Remote Sites Dial In and Dial Out

3. Describe the central and remote site configurations for three setups: central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Also describes the usage of Point-to-Point Protocol (PPP) encapsulation and the Challenge Handshake Authentication Protocol (CHAP).

• Having Remote Sites Dial Out

A common configuration is one in which the remote sites place calls to the central site but the central site does not dial out. In a "star" topology, it is possible for all of the remote routers to have their serial interfaces on the same subnet as the central site serial interface.

• Using DDR as a Backup to Leased Lines

Describes the use of DDR as a backup method to leased lines and provides examples of how to use floating static routes on single and shared interfaces.

• Using Leased Lines and Dial Backup

Describes the use of Data Terminal Ready (DTR) dialing and V.25bis dialing with leased lines.

Case Study 4:

Network Security

This case study should provide the specific actions you can take to improve the security of your network. Before going into specifics, however, you should understand the following basic concepts that are essential to any security system:

• Know your enemy

This case study refers to attackers or intruders. Consider who might want to circumvent

4. your security measures and identify their motivations. Determine what they might want to do and the damage that they could cause to your network. Security measures can never make it impossible for a user to perform unauthorized tasks with a computer system. They can only make it harder. The goal is to make sure the network security controls are beyond the attacker's ability or motivation.

• Count the cost

Security measures almost always reduce convenience, especially for sophisticated users. Security can delay work and create expensive administrative and educational overhead. It

can use significant computing resources and require dedicated hardware. When you design your security measures, understand their costs and weigh those costs against the potential benefits. To do that, you must understand the costs of the measures themselves and the costs and likelihoods of security breaches. If you incur security costs out of proportion to the actual dangers, you have done yourself a disservice.

• Identify your assumptions

Every security system has underlying assumptions. For example, you might assume that your network is not tapped, or that attackers know less than you do, that they are using standard software, or that a locked room is safe. Be sure to examine and justify your assumptions. Any hidden assumption is a potential security hole.

• Control your secrets

Most security is based on secrets. Passwords and encryption keys, for example, are secrets. Too often, though, the secrets are not really all that secret. The most important part of keeping secrets is knowing the areas you need to protect. What knowledge would enable someone to circumvent your system? You should jealously guard that knowledge and assume that everything else is known to your adversaries. The more secrets you have, the harder it will be to keep all of them. Security systems should be designed so that only a limited number of secrets need to be kept.

 Know your weaknesses

Every security system has vulnerabilities. You should understand your system's weak points and know how they could be exploited. You should also know the areas that present the largest danger and prevent access to them immediately. Understanding the weak points is the first step toward turning them into secure areas.

 Limit the scope of access

You should create appropriate barriers inside your system so that if intruders access one part of the system, they do not automatically have access to the rest of the system. The security of a system is only as good as the weakest security level of any single host in the system.

• Remember physical security Physical access to a computer (or a router) usually gives a sufficiently sophisticated user total control over that computer. Physical access to a network link usually allows a person to tap that link, jam it, or inject traffic into it. It makes no sense to install complicated software security measures when access to the hardware is not controlled.

Case Study 5:

Controlling Traffic Flow

5. In this case study, the firewall router allows incoming new connections to one or more communication servers or hosts. Having a designated router act as a firewall is desirable because it clearly identifies the router's purpose as the external gateway and avoids encumbering other routers with this task. In the event that the internal network needs to isolate itself, the firewall router provides the point of isolation so that the rest of the internal network structure is not affected. Connections to the hosts are restricted to incoming file transfer protocol (FTP) requests and email services. The incoming Telnet, or modem connections to the communication server are screened by the communication server running TACACS username authentication.

Case Study 6:

Defining Access Lists

6. Access lists define the actual traffic that will be permitted or denied, whereas an access group applies an access list definition to an interface. Access lists can be used to deny connections that are known to be a security risk and then permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, the latter is the more secure method. In this case study, incoming email and news are permitted for a few hosts, but FTP, Telnet, and rlogin services are permitted only to hosts on the firewall subnet. IP extended access lists (range 100 to 199) and

transmission control protocol (TCP) or user datagram protocol (UDP) port numbers are used to filter traffic. When a connection is to be established for email, Telnet, FTP, and so forth, the connection will attempt to open a service on a specified port number. You can, therefore, filter out selected types of connections by denying packets that are attempting to use that service. An access list is invoked after a routing decision has been made but before the packet is sent out on an interface. The best place to define an access list is on a preferred host using your favorite text editor. You can create a file that contains
the access-list commands, place the file (marked readable) in the default TFTP directory, and then network load the file onto the router.

7. Case Study 7:
Configuring a fire wall
Consider a Fire wall communication server with single inbound modem. Configure the modem to ensure security for LAN

8. Case Study 8:
Integrating EIGRP (Enhanced Interior Gateway Routing Protocol) into Existing Networks:
The case study should provide the benefits and considerations involved in integrating Enhanced IGRP into the following types of internetworks:
• IP—The existing IP network is running IGRP
• Novell IPX—The existing IPX network is running RIP and SAP
• AppleTalk—The existing AppleTalk network is running the Routing Table Maintenance Protocol (RTMP)
When integrating Enhanced IGRP into existing networks, plan a phased implementation. Add Enhanced IGRP at the periphery of the network by configuring Enhanced IGRP on a boundary router on the backbone off the core network. Then integrate Enhanced IGRP into the core network.

Total Practical Hours **45**

Course Outcome

CO1: Design and apply various configurations and protocols in LAN.
CO2: Design and implement RIP and OSPF Redistribution and Dial-on-Demand Routing.
CO3: Design and implement Network Security.
CO4: Design Controlling Traffic Flow and Defining Access Lists.
CO5: Implement Configuring a fire wall and Integrating EIGRP into Existing Networks.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2201	RESEARCH METHODOLOGY	3	1	0	4

Course Objective

1. To develop understanding of the basic framework of research process.
2. To develop an understanding of various research designs and techniques.
3. To identify various sources of information for literature review and data collection.
4. To develop an understanding of the ethical dimensions of conducting applied research.
5. Appreciate the components of scholarly writing and evaluate its quality

Unit	Description	Instructional Hours
I	**OBJECTIVES AND TYPES OF RESEARCH** Motivation and objectives – Research methods vs Methodology. Types of research – Descriptive vs. Analytical, Applied vs Fundamental, Quantitative vs. Qualitative, Conceptual vs Empirical.	12
II	**RESEARCH FORMULATION** Defining and formulating the research problem - Selecting the problem - Necessity of defining the problem - Importance of literature review in defining a problem – Literature review – Primary and secondary sources – reviews, treatise, monographs-patents – web as a source – searching the web - Critical literature review – Identifying gap areas from literature review - Development of working hypothesis.	12
III	**RESEARCH DESIGN AND METHODS** Research design – Basic Principles- Need of research design — Features of good design – Important concepts relating to research design – Observation and Facts, Laws and Theories, Prediction and explanation, Induction, Deduction, Development of Models. Developing a research plan - Exploration, Description, Diagnosis, Experimentation. Determining experimental and sample designs.	12
IV	**DATA COLLECTION AND ANALYSIS** Execution of the research - Observation and Collection of data - Methods of data collection – Sampling Methods- Data Processing and Analysis strategies - Data Analysis with Statistical Packages - Hypothesis-testing - Generalization and Interpretation.	12
V	**REPORTING AND THESIS WRITING** Structure and components of scientific reports - Types of report – Technical reports and thesis – Significance – Different steps in the preparation – Layout, structure and Language of typical reports – Illustrations and tables - Bibliography, referencing and footnotes - Oral presentation – Planning – Preparation – Practice – Making presentation – Use of visual aids - Importance of effective communication	12
	Total Instructional Hours	**60**

Course Outcome

CO1: To understand some basic concepts of research and its methodologies
CO2: To identify appropriate research topics
CO3: To select and define appropriate research problem and parameters
CO4: To prepare a project proposal (to undertake a project)
CO5: To write a research report and thesis

REFERENCE BOOKS :

R1 - Garg, B.L., Karadia, R., Agarwal, F. and Agarwal, U.K., 2002. An introduction to Research Methodology, RBSA Publishers.
R2 - Kothari, C.R., 2004. Research Methodology: Methods and Techniques. New Age International.
R3 - Sinha, S.C. and Dhiman, A.K., 2002. Research Methodology, Ess Ess Publications. 2 volumes.
R4 - Trochim, W.M.K., 2005. Research Methods: the concise knowledge base, Atomic Dog Publishing.
R5 - Ranjit Kumar, 2011, RESEARCH METHODOLOGY- a step-by-step guide for beginners, SAGE Publications.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2202	ADVANCED DATABASES	3	0	0	3

Course Objective

1. To learn the modeling and design of databases.
2. To acquire knowledge on parallel and distributed databases and its applications.
3. To study the usage and applications of Object Oriented database
4. To understand the principles of intelligent databases and usage of advanced data models.
5. To learn emerging databases such as XML, Cloud and Big Data and acquire inquisitive attitude towards research topics in databases.

Unit	Description	Instructional Hours
I	**PARALLEL AND DISTRIBUTED DATABASES** Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems- Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Case Studies	9
II	**OBJECT AND OBJECT RELATIONAL DATABASES** Concepts for Object Databases: Object Identity – Object structure – Type Constructors – Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies – Inheritance – Complex Objects – Object Database Standards, Languages and Design: ODMG Model – ODL – OQL – Object Relational and Extended – Relational Systems: Object Relational features in SQL/Oracle – Case Studies.	9
III	**INTELLIGENT DATABASES** Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy- Applications- Design Principles for Active Rules- Temporal Databases: Overview of Temporal Databases- TSQL2- Deductive Databases: Logic of Query Languages – Datalog- Recursive Rules-Syntax and Semantics of Datalog Languages- Implementation of Rules and Recursion- Recursive Queries in SQL- Spatial Databases- Spatial Data Types- Spatial Relationships- Spatial Data Structures- Spatial Access Methods- Spatial DB Implementation.	9
IV	**ADVANCED DATA MODELS** Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models -Concurrency Control - Transaction Commit Protocols- Multimedia Databases- Information Retrieval- Data Warehousing- Data Mining- Text Mining.	9
V	**EMERGING TECHNOLOGIES** XML Databases: XML-Related Technologies-XML Schema- XML Query Languages- Storing XML in Databases-XML and SQL- Native XML Databases- Web Databases- Geographic Information Systems- Biological Data Management- Cloud Based Databases: Data Storage Systems on the Cloud- Cloud Storage Architectures-Cloud Data Models- Query Languages- Introduction to Big Data-Storage-Analysis.	9
	Total Instructional Hours	**45**

Course Outcome

CO1; Select the appropriate high performance database like parallel and distributed database
CO2: Model and represent the real world data using object oriented database
CO3: Design a semantic based database to meaningful data access and Embed the rule set in the database to implement intelligent databases
CO4: Represent the data using XML database for better interoperability
CO5: Handle Big data and store in a transparent manner in the cloud and solve the issues related to the data storage and retrieval

REFERENCE BOOKS :

R1- R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, Fifth Edition, Pearson Education/Addison Wesley, 2007.

R2- Thomas Cannolly and Carolyn Begg, “Database Systems, A Practical Approach to Design, Implementation and Management”, Third Edition, Pearson Education, 2007.

R3- Henry F Korth, Abraham Silberschatz, S. Sudharshan, “Database SystemConcepts”, Fifth Edition, McGraw Hill, 2006.

R4- C.J.Date, A.Kannan and S.Swamynathan, ”An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

R5- Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, McGraw Hill, Third Edition 2004.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2203	PRINCIPLES OF PROGRAMMING LANGUAGES	3	0	0	3

Course Objective

1. To understand and describe syntax and semantics of programming languages
2. To understand data, data types, and basic statements
3. To understand call-return architecture and ways of implementing them
4. To understand object-orientation, concurrency, and event handling in programming languages
5. To develop programs in non-procedural programming paradigms

Unit	Description	Instructional Hours
I	**SYNTAX AND SEMANTICS** Evolution of programming languages – describing syntax – context-free grammars – attribute grammars – describing semantics – lexical analysis – parsing – recursive-decent – bottom-up parsing	9
II	**DATA, DATA TYPES, AND BASIC STATEMENTS** Names – variables – binding – type checking – scope – scope rules – lifetime and garbage collection – primitive data types – strings – array types – associative arrays – record types – union types – pointers and references – Arithmetic expressions – overloaded operators – type conversions – relational and boolean expressions – assignment statements – mixed-mode assignments – control structures – selection – iterations – branching – guarded statements	9
III	**SUBPROGRAMS AND IMPLEMENTATIONS** Subprograms – design issues – local referencing – parameter passing – overloaded methods – generic methods – design issues for functions – semantics of call and return – implementing simple subprograms – stack and dynamic local variables – nested subprograms – blocks – dynamic scoping	9
IV	**OBJECT-ORIENTATION, CONCURRENCY, AND EVENT HANDLING** Object-orientation – design issues for OOP languages – implementation of object-oriented constructs – concurrency – semaphores – monitors – message passing – threads – statement level concurrency – exception handling – even handling	9
V	**FUNCTIONAL AND LOGIC PROGRAMMING LANGUAGES** Introduction to lambda calculus – fundamentals of functional programming languages – Programming with Scheme – Programming with ML – Introduction to logic and logic programming – Programming with Prolog – multi-paradigm languages	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Describe syntax and semantics of programming languages
CO2: Explain data, data types, and basic statements of programming languages
CO3: Design and implement subprogram constructs and apply object-oriented, concurrency, and event handling programming constructs
CO4: Develop programs in Scheme, ML, and Prolog
CO5: Understand and adopt new programming languages

REFERENCE BOOKS :

R1- Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
R2- Michael L. Scott, "Programming Language Pragmatics", Third Edition, Morgan Kaufmann, 2009.
R3- R. Kent Dybvig "The Scheme programming language", Fourth Edition, MIT Press, 2009.
R4- Jeffrey D. Ullman, "Elements of ML programming", Second Edition, Prentice Hall, 1998.
R5- Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009.
R6- W. F. Clocksin and C. S. Mellish, "Programming in Prolog: Using the ISO Standard", Fifth Edition, Springer, 2003.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2204	ADVANCED OPERATING SYSTEMS	3	0	0	3

Course Objective

1. To learn the fundamentals of Operating Systems
2. To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
3. To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols
4. To know the components and management aspects of Real time, Mobile operating systems
5. To make case study about linux system.

.Unit	Description	Instructional Hours
I	**FUNDAMENTALS OF OPERATING SYSTEMS** Overview – Synchronization Mechanisms – Processes and Threads - Process Scheduling – Deadlocks: Detection, Prevention and Recovery – Models of Resources – Memory Management Techniques.	9
II	**DISTRIBUTED OPERATING SYSTEMS** Issues in Distributed Operating System – Architecture – Communication Primitives – Lamport's Logical clocks – Causal Ordering of Messages – Distributed Mutual Exclusion Algorithms – Centralized and Distributed Deadlock Detection Algorithms – Agreement Protocols.	9
III	**DISTRIBUTED RESOURCE MANAGEMENT** Distributed File Systems – Design Issues - Distributed Shared Memory – Algorithms for Implementing Distributed Shared memory–Issues in Load Distributing – Scheduling Algorithms – Synchronous and Asynchronous Check Pointing and Recovery – Fault Tolerance – Two-Phase Commit Protocol – Nonblocking Commit Protocol – Security and Protection.	9
IV	**REAL TIME AND MOBILE OPERATING SYSTEMS** Basic Model of Real Time Systems - Characteristics- Applications of Real Time Systems – Real Time Task Scheduling - Handling Resource Sharing - Mobile Operating Systems –Micro Kernel Design - Client Server Resource Access – Processes and Threads - Memory Management – File system.	9
V	**CASE STUDIES** Linux System: Design Principles - Kernel Modules - Process Management Scheduling – Memory Management - Input-Output Management - File System - Interprocess Communication. iOS and Android: Architecture and SDK Framework - Media Layer - Services Layer - Core OS Layer – File System.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Discuss the various synchronization, scheduling and memory management issues
CO2: Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
CO3: Discuss the various resource management techniques for distributed systems
CO4: Identify the different features of real time and mobile operating systems and install and use available open source kernel
CO5: Modify existing open source kernels in terms of functionality or features used

REFERENCE BOOKS :

R1-Mukesh Singhal and Niranjan G. Shivaratri, "Advanced Concepts in Operating Systems – Distributed, Database, and Multiprocessor Operating Systems", Tata McGraw-Hill, 2001.
R2- Abraham Silberschatz; Peter Baer Galvin; Greg Gagne, "Operating System Concepts", Seventh Edition, John Wiley & Sons, 2004.
R3- Daniel P Bovet and Marco Cesati, "Understanding the Linux kernel", 3rd edition, O'Reilly, 2005.
R4- Rajib Mall, "Real-Time Systems: Theory and Practice", Pearson Education India, 2006.
R5- Neil Smyth, "iPhone iOS 4 Development Essentials – Xcode", Fourth Edition, Payload media, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2001	ADVANCED DATABASE LABORATORY	0	0	4	2

Course Objective

1. To learn to work on distributed data bases
2. To understand and work on object oriented databases and to gain knowledge in parallel data base by experimenting it
3. To learn to work on active database and to study and explore deductive database
4. To work on the data mining tool *weka*
5. To represent and work with the database using XML

Expt. No. **Description of the Experiments**

1.

DISTRIBUTED DATABASE:

1. Consider a distributed database for a bookstore with 4 sites called S1, S2, S3 and S4.
Consider the following relations:
Books (ISBN, primary Author, topic, total Stock, price)
Book Store (store No, city, state, zip, inventoryValue)
Stock (store No, ISBN, Qty)
Total Stock is the total number of books in stock and inventory Value is the total inventory
value for the store in dollars.
Consider that Books are fragmented by price amounts into:19
F1: Books: price up to $20
F2: Books: price from $20.01 to $50
F3: Books: price from $50.01 to $100
F4: Books: price $100.01 and above
Similarly, Book Stores are divided by ZIP codes into:
S1: Bookstore: Zip up to 25000
S2: Bookstore: Zip 25001 to 50000
S3: Bookstore: Zip 50001 to 75000
S4: Bookstore: Zip 75001 to 99999
Task: Write SQL query for the following
1. Insert and Display details in each table.
2. Find the total number of books in stock where price is between $15 and $55.
3. Update the book price of book No=1234 from $45 to $55 at site S3.
4. Find total number of book at site S2.

2. Implement deadlock detection algorithm for distributed database using wait-for graph and
test with the following information.
Consider five transactions T1, T2, T3, T4 and T5 with
T1 initiated at site S1 and spawning an agent at site S2
T2 initiated at site S3 and spawning an agent at site S1
T3 initiated at site S1 and spawning an agent at site S3
T4 initiated at site S2 and spawning an agent at site S3
T5 initiated at site S3
The locking information for these transactions is shown in the following table
Transactions Data items locked by transactions Data items transaction is waiting for
Site involved in operations
T1 X1 X8 S1
T1 X6 X2 S2
T2 X4 X1 S1
T2 X5 - S3
T3 X2 X7 S1
T3 - X3 S3

T4 X7 - S2
T4 X8 X5 S3
T5 X3 X7 S3
Produce local wait for graph for each of the sites and construct global wait for graph and check for dead lock.

OBJECT ORIENTED DATABASE:

3. A University wants to track persons associated with them. A person can be an Employee or Student. Employees are Faculty, Technicians and Project associates. Students are Full time students, Part time students and Teaching Assistants.

a) Design an Enhanced Entity Relationship (EER) Model for university database. Write OQL for the following

i. Insert details in each object.
ii. Display the Employee details.
iii. Display Student Details.
iv. Modify person details.

2. v. Delete person details.

b) Extend the design by incorporating the following information.

20 Students are registering for courses which are handled by instructor researchers (graduate students). Faculty are advisors to graduate students. Instructor researchers' class is a category with super class of faculty and graduate students. Faculty are having sponsored research projects with a grant supporting instruction researchers. Grants are sanctioned by different agencies.

Faculty belongs to different departments. Department is chaired by a faculty. Implement for the

Insertion and Display of details in each class.

PARALLEL DATABASE:

4. Consider the application for University Counselling for Engineering Colleges. The college, department and vacancy details are maintained in 3 sites. Students are allocated colleges in these 3 sites simultaneously. Implement this application using parallel database [State any assumptions you have made].

3. 5. There are 5 processors working in a parallel environment and producing output. The output record contains college details and students mark information. Implement parallel join and parallel sort algorithms to get the marks from different colleges of the university and publish 10 ranks for each discipline.

ACTIVE DATABASE:

6. Create triggers and assertions for Bank database handling deposits and loan and admission database handling seat allocation and vacancy position. Design the above relational database schema and implement the following triggers and assertions.

a. When a deposit is made by a customer, create a trigger for updating customers account and bank account

4. b. When a loan is issued to the customer, create a trigger for updating customer's loan account and bank account.

c. Create assertion for bank database so that the total loan amount does not exceed the total balance in the bank.

d. When an admission is made, create a trigger for updating the seat allocation details and vacancy position.

DEDUCTIVE DATABASE:

5. 7. Construct a knowledge database for kinship domain (family relations) with facts. Extract the following relations using rules.

Parent, Sibling, Brother, Sister, Child, Daughter, Son, Spouse, Wife, husband,

Grandparent, Grandchild, Cousin, Aunt and Uncle.

WEKA TOOL:

8. Work with Weka tool classification and clustering algorithms using the given training data and test with the unknown sample. Also experiment with different scenarios and large data set

RID Age Income Student Credit_
rating
Class: buys_
computer
1 youth high no fair no
2 youth high no excellent no
3 middle_aged high no fair yes
4 senior medium no fair yes
6. 5 senior low yes fair yes
6 senior low yes excellent no
7 middle_aged low yes excellent yes
8 youth medium no fair no
9 youth low yes fair yes
10 senior medium yes fair yes
11 Youth medium yes excellent yes
12 middle_aged medium no excellent yes
21
13 middle_aged high yes fair yes
14 senior medium no excellent no

QUERY PROCESSING

9. Implement Query Optimizer with Relational Algebraic expression construction and execution plan generation for choosing an efficient execution strategy for processing the given query.

7. Also design employee database and test the algorithm with following sample queries.

a) Select empid, empname from employee where experience > 5
b) Find all managers working at London Branch

XML

10. Design XML Schema for the given company database

Department (deptName, deptNo, deptManagerSSN, deptManagerStartDate, deptLocation)

Employee (empName, empSSN, empSex, empSalary, empBirthDate, empDeptNo, empSupervisorSSN, empAddress, empWorksOn)

Project (projName, projNo, projLocation, projDeptNo, projWorker)

a. Implement the following queries using XQuery and XPath

i. Retrieve the department name, manager name, and manager salary for every department'

8. ii. Retrieve the employee name, supervisor name and employee salary for each employee who works in the Research Department.

iii. Retrieve the project name, controlling department name, number of employees and total hours worked per week on the project for each project.

iv. Retrieve the project name, controlling department name, number of employees and total hours worked per week on the project for each project with more than one employee working on it

b. Implement a storage structure for storing XML database and test with the above schema.

Total Practical Hours	**45**

Course Outcome

CO1: Work on distributed databases
CO2: Create and work on object oriented databases and parallel database
CO3: Experiment on active database and explore the features of deductive database
CO4: To work on weka tool for clustering and classification
CO5: Represent the database using XML and work on it

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2702	CASE STUDY – OPERATING SYSTEMS DESIGN (TEAM WORK)	0	0	4	2

Course Objective

1. To develop capabilities to work at systems level
2. To learn about issues in designing and implementing modern operating systems
3. To understand team formation, team issues, and allocating roles and responsibilities
4. To make effective presentations on the work done
5. To develop effective written communication skills

Expt. No.	Description of the Experiments	
1.	A team of three or four students will work on assigned case study / mini-project. Case Study / Mini-project can be designed on the following lines: 1. Development of a reasonably sized dynamically loadable kernel module for Linux kernel 2. Study educational operating systems such as Minix (http://www.minix3.org/), Weenix (http://weenix.cs.brown.edu/mediawiki/index.php/Weenix) and develop reasonably sized interesting modules for them 3. Study the Android open source operating system for mobile devices (http://source.android.com/) and develop / modify some modules. 4. Study any embedded and real-time operating system such as eCos (http://ecos.sourceware.org/) and develop / modify some modules.	60
	Total Practical Hours	**60**

Course Outcome

CO1; Develop assigned modules of operating systems design carrying out coding, testing, and documentation work involved.
CO2: Describe team issues and apply suitable methods to resolve the same.
CO3: Demonstrate individual competence in building medium size operating system components and ethical and professional attributes of a computer engineer.
CO4: Prepare suitable plan with clear statements of deliverables, and track the same.
CO5: Make individual presentation of the work carried out and prepare well-organized written documents to communicate individual work accomplished.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP3201	SOFTWARE PROCESS AND PROJECT MANAGEMENT	3	1	0	4

Course Objective

1. To understand overall SDLC and adopt suitable processes
2. To elicite, analyze, prioritize, and manage both functional and quality requirements
3. To estimate efforts required, plan, and track the plans
4. To understand and apply configuration and quality management techniques
5. To evaluate, manage, and design processes

Unit	Description	Instructional Hours
I	**DEVELOPMENT LIFE CYCLE PROCESSES** Overview of software development life cycle – introduction to processes – Personal Software Process (PSP) – Team software process (TSP) – Unified processes – agile processes – choosing the right process **Tutorial:** Software development using PSP	9+3
II	**REQUIREMENTS MANAGEMENT** Functional requirements and quality attributes – elicitation techniques – Quality Attribute Workshops (QAW) – analysis, prioritization, and trade-off – Architecture Centric Development Method (ACDM) – requirements documentation and specification – change management – traceability of requirements **Tutorial:** Conduct QAW, elicit, analyze, prioritize, and document requirements using ACDM	9+3
III	**ESTIMATION, PLANNING, AND TRACKING** Identifying and prioritizing risks – risk mitigation plans – estimation techniques – use case points – function points – COCOMO II – top-down estimation – bottom-up estimation – work breakdown structure – macro and micro plans – planning poker – wideband delphi – documenting the plan – tracking the plan – earned value method (EVM) **Tutorial:** Estimation, planning, and tracking exercises	9+3
IV	**CONFIGURATION AND QUALITY MANAGEMENT** identifying artifacts to be configured – naming conventions and version control – configuration control – quality assurance techniques – peer reviews – Fegan inspection – unit, integration, system, and acceptance testing – test data and test cases – bug tracking – causal analysis **Tutorial:** version control exercises, development of test cases, causal analysis of defects	9+3
V	**SOFTWARE PROCESS DEFINITION AND MANAGEMENT** Process elements – process architecture – relationship between elements – process modeling – process definition techniques – ETVX (entry-task-validation-exit) – process baselining – process assessment and improvement – CMMI – Six Sigma **Tutorial:** process measurement exercises, process definition using ETVX	9+3
	Total Instructional Hours	**60**

Course Outcome

CO1: Explain software development life cycle and adopt a suitable process for software development
CO2: Elicit functional and quality requirements, analyze, prioritize, and manage requirements and perform trade-off among conflicting requirements
CO3: Identify and prioritize risks and create mitigation plans, estimate the efforts required for software development and perform planning and tracking activities
CO4: Control the artifacts during software development and perform various tests to ensure quality
CO5: Define new processes based on the needs and adopt best practices for process improvement

REFERENCE BOOKS :

R1 - Pankaj Jalote, "Software Project Management in Practice", Pearson, 2002.
R2 - Chris F. Kemerer, "Software Project Management – Readings and Cases", McGraw Hill, 1997.
R3 - Watts S. Humphrey, "PSP: A self-improvement process for software engineers", Addison- Wesley, 2005.
R4 - Watts S. Humphrey, "Introduction to the Team Software Process", Addison-Wesley, 2000.
R5 - Orit Hazzan and Yael Dubinsky, "Agile software engineering", Springer, 2008.
R6 - James R. Persse, "Process Improvement Essentials", O'Reilly, 2006.
R7 - Roger S. Pressman, "Software Engineering – A Practitioner's Approach", Seventh Edition, McGraw Hill, 2010.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3202	ADVANCED INTERNET OF THINGS	3	0	0	3

Course Objective

1. To understand the basics of Internet of Things and to understand the middleware for Internet of Things
2. To understand the IOT protocols
3. To understand the concepts of Web of Things
4. To understand the concepts of different models for network dynamics
5. To understand the concepts of IoT applications

Unit	Description	Instructional Hours
I	**INTRODUCTION** Definitions and Functional Requirements –Motivation – Architecture - Web 3.0 View of IoT– Ubiquitous IoT Applications – Four Pillars of IoT – DNA of IoT - The Toolkit Approach for End-user Participation in the Internet of Things. Middleware for IoT: Overview – Communication middleware for IoT –IoT Information Security	9
II	**IOT PROTOCOLS** Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE 802.15.4 – BACNet Protocol – Modbus – KNX – Zigbee Architecture – Network layer – APS layer – Security	9
III	**WEB OF THINGS** Web of Things versus Internet of Things – Two Pillars of the Web – Architecture Standardization for WoT– Platform Middleware for WoT – Unified Multitier WoT Architecture – WoT Portals and Business Intelligence. Cloud of Things: Grid/SOA and Cloud Computing – Cloud Middleware – Cloud Standards – Cloud Providers and Systems – Mobile Cloud Computing – The Cloud of Things Architecture	9
IV	**INTEGRATED MODELS** Integrated Billing Solutions in the Internet of Things Business Models for the Internet of Things - Network Dynamics: Population Models – Information Cascades - Network Effects - Network Dynamics: Structural Models - Cascading Behavior in Networks - The Small-World Phenomenon	9
V	**APPLICATIONS** The Role of the Internet of Things for Increased Autonomy and Agility in Collaborative Production Environments - Resource Management in the Internet of Things: Clustering, Synchronization and Software Agents. Applications - Smart Grid – Electrical Vehicle Charging	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify and design the new models for market strategic interaction and Design a middleware for IoT
CO2: Analyze various protocols for IoT
CO3: Design business intelligence and information security for WoB
CO4: Analyze and design different models for network dynamics
CO5: Design IoT applications – Smart Grid – Electrical Vehicle Charging

REFERENCE BOOKS :

R1 - The Internet of Things in the Cloud: A Middleware Perspective - Honbo Zhou – CRC Press – 2012
R2 - Architecting the Internet of Things - Dieter Uckelmann; Mark Harrison; Florian Michahelles- (Eds.) – Springer – 2011
R3 - Networks, Crowds, and Markets: Reasoning About a Highly Connected World - David Easley and Jon Kleinberg, Cambridge University Press - 2010
R4 - The Internet of Things: Applications to the Smart Grid and Building Automation by - Olivier Hersent, Omar Elloumi and David Boswarthick - Wiley -2012
R5 - Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Key applications and Protocols", Wiley, 2012

ELECTIVE - I

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1301	DATA ANALYSIS AND BUSINESS INTELLIGENCE	3	0	0	3

Course Objective

1. To understand linear regression models and logistic regression models
2. To understand generalized linear models
3. To understand simulation using regression models
4. To understand causal inference and multilevel regression
5. To understand data collection and model understanding

Unit	Description	Instructional Hours
I	**LINEAR REGRESSION** Introduction to data analysis – Statistical processes – statistical models –review of random variables and probability distributions – linear regression – one predictor – multiple predictors –Interactions statistical Inference- prediction and validation – linear transformations – centering and standardizing – correlation – logarithmic transformations – other transformations – building regression models – fitting a series of regressions	9
II	**LOGISTIC AND GENERALIZED LINEAR MODELS** Logistic regression – logistic regression coefficients – latent-data formulation – building a logistic regression model – logistic regression with interactions – evaluating, checking, and comparing fitted logistic regressions – identifiability and separation – Poisson regression – logistic-binomial model – Probit regression – multinomial regression – robust regression using *t* model – building complex generalized linear models – constructive choice models	9
III	**SIMULATION AND CAUSAL INFERENCE** Simulation of probability models – summarizing linear regressions – simulation of non-linear predictions – predictive simulation for generalized linear models – fake-data simulation –simulating and comparing to actual data – predictive simulation to check the fit of a time-series model – causal inference – randomized experiments and predictive compositions– observational studies	9
IV	**MULTILEVEL REGRESSION** Multilevel structures – clustered data –costs and benefits of multilevel modeling- multilevel linear models – partial pooling –quickly filling-multilevel models in R- group-level predictors – model building and statistical significance – varying intercepts and slopes – non-nested models – multi-level logistic regression – multi-level generalized linear models	9
V	**DATA COLLECTION AND MODEL UNDERSTANDING** Design of data collection – classical power calculations – multilevel power calculations – power calculation using fake-data simulation – understanding and summarizing fitted models –Regression-straight line regression-multiple linear regression-analysis of variance-sum of square and R^2 – multiple comparisons and statistical significance – analysis of variance – ANOVA and multilevel linear and general linear models – missing data imputation	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Build and apply linear regression models and logistic regression models
CO2: Build and apply generalized linear models
CO3: Perform simulation using regression models
CO4: Perform casual inference from data
CO5: Build and apply multilevel regression models and perform data collection and variance analysis

REFERENCE BOOKS :

R1- Andrew Gelman and Jennifer Hill, "Data Analysis using Regression and multilevel / hierarchical multilevel/Hierarchical Models", Cambridge University Press, 2006.

R2- Philipp K. Janert, "Data Analysis with Open Source Tools", O'Reilley, 2010.

R3- Wes McKinney, "Python for Data Analysis", O'Reilley, 2012.

R4- Davinderjit Sivia and John Skilling, "Data Analysis: A Bayesian Tutorial", Second Edition, Oxford University Press, 2006.

R5- Robert Nisbelt, John Elder, and Gary Miner, "Handbook of statistical analysis and data mining applications", Academic Press, 2009.

R6- Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, " Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.

R7- John Maindonald and W. John Braun, "Data Analysis and Graphics Using R: An Examplebased Approach", Third Edition, Cambridge University Press, 2010.

R8- David Ruppert, "Statistics and Data Analysis for Financial Engineering", Springer, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1302	FORMAL MODELS OF SOFTWARE SYSTEMS	3	0	0	3

Course Objective

1. To understand the basic elements of Z
2. To understand relations, functions, and logical structures in Z
3. To understand Z schemas and schema calculus
4. To learn selected Z case studies
5. To understand Z schema refinement

Unit	Description	Instructional Hours
I	**FOUNDATIONS OF Z** Understanding formal methods – motivation for formal methods – informal requirements to formalspecifications – validating formal specifications – Overview of Z specification – basic elements of Z– sets and types – declarations – variables – expressions – operators – predicates and equations	9
II	**STRUCTURES IN Z** Tuples and records – relations, tables, databases – pairs and binary relations – functions – sequences – propositional logic in Z – predicate logic in Z – Z and boolean types – set comprehension – lambda calculus in Z – simple formal specifications – modeling systems and change	9
III	**Z SCHEMAS AND SCHEMA CALCULUS** Z schemas – schema calculus – schema conjunction and disjunction – other schema calculus operators – schema types and bindings – generic definitions – free types – formal reasoning – checking specifications – precondition calculation – machine-checked proofs	9
IV	**Z CASE STUDIES** Case Study: Text processing system – Case Study: Eight Queens – Case Study: Graphical User Interface – Case Study: Safety critical protection system – Case Study: Concurrency and real time systems	9
V	**Z REFINEMENT** Refinement of Z specification – generalizing refinements – refinement strategies – program derivation and verification – refinement calculus – data structures – state schemas – functions and relations – operation schemas – schema expressions – refinement case study	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Apply the basic elements of Z
CO2: Develop relational, functional, and logical Z structures
CO3: Develop Z schema as models of software systems
CO4: Perform verifications and conduct proofs using Z models
CO5: Refine Z models towards implementing software systems

REFERENCE BOOKS :

R1- Jonathan Jacky, "The way of Z: Practical programming with formal methods", Cambridge University Press, 1996.
R2- Antoni Diller, "Z: An introduction to formal methods", Second Edition, Wiley, 1994.
R3- Jim Woodcock and Jim Davies, "Using Z – Specification, Refinement, and Proof", Prentice Hall, 1996.
R4- J. M. Spivey, "The Z notation: A reference manual", Second Edition, Prentice Hall, 1992.
R5- M. Ben-Ari, "Mathematical logic for computer science", Second Edition, Springer, 2003.
R6- M. Huth and M. Ryan, "Logic in Computer Science – Modeling and Reasoning about systems", Second Edition, Cambridge University Press, 2004.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1303	PERFORMANCE EVALUATION OF COMPUTER SYSTEMS	3	0	0	3

Course Objective

1. To understand the mathematical foundations needed for performance evaluation of computer systems
2. To understand the metrics used for performance evaluation
3. To understand the analytical modeling of computer systems
4. To enable the students to develop new queuing analysis for both simple and complex systems
5. To appreciate the use of smart scheduling and introduce the students to analytical techniques for evaluating scheduling policies

Unit	Description	Instructional Hours
I	**OVERVIEW OF PERFORMANCE EVALUATION** Need for Performance Evaluation in Computer Systems – Overview of Performance Evaluation Methods – Introduction to Queueing – Probability Review – Generating Random Variables for Simulation – Sample Paths, Convergence and Averages – Little's Law and other Operational Laws – Modification for Closed Systems.	9
II	**MARKOV CHAINS AND SIMPLE QUEUES** Discrete-Time Markov Chains – Ergodicity Theory – Real World Examples – Google, Aloha – Transition to Continuous-Time Markov Chain – M/M/1 and PASTA.	9
III	**MULTI-SERVER AND MULTI-QUEUE SYSTEMS** Server Farms: M/M/k and M/M/k/k – Capacity Provisioning for Server Farms – Time Reversibility and Burke's Theorem – Networks of Queues and Jackson Product Form – Classed and Closed Networks of Queues.	9
IV	**REAL-WORLD WORKLOADS** Case Study of Real-world Workloads – Phase-Type Distributions and Matrix-Alalytic Methods – Networks with Time-Sharing Servers – M/G/1 Queue and the Inspection Paradox – Task Assignment Policies for Server Farms.	9
V	**SMART SCHEDULING IN THE M/G/1** Performance Metrics – Scheduling Non-Preemptive and Preemptive Non-Size-Based Policies - Scheduling Non-Preemptive and Preemptive Size-Based Policies – Scheduling - SRPT and Fairness.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the need for performance evaluation and the metrics used for it
CO2: Discuss open and closed queueing networks
CO3: Define Little'e law and other operational laws
CO4: Apply the operational laws to open and closed systems
CO5: Use discrete-time and continuous-time Markov chains to model real world systems and develop analytical techniques for evaluating scheduling policies

REFERENCE BOOKS :

R1- Mor Harchol - Balter, "Performance Modeling and Design of Computer Systems – Queueing Theory in Action", Cambridge University Press, 2013.

R2- Raj Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling”, Wiley-Interscience, 1991.
R3- Lieven Eeckhout, “Computer Architecture Performance Evaluation Methods”, Morgan and Claypool Publishers, 2010.
R4- Paul J. Fortier and Howard E. Michel, “Computer Systems Performance Evaluation and Prediction”, Elsevier, 2003.
R5- David J. Lilja, “Measuring Computer Performance: A Practitioner’s Guide”, Cambridge University Press, 2000.
R6- Krishna Kant, “Introduction to Computer System Performance Evaluation”, McGraw-Hill, 1992.
R7- K. S. Trivedi, “Probability and Statistics with Reliability, Queueing and Computer Science Applications”, John Wiley and Sons, 2001.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1304	PROBABILISTIC REASONING SYSTEMS	3	0	0	3

Course Objective

1. To construct and reason with Bayesian networks
2. To reason with temporal models
3. To make exact and approximate inferences with graphical models
4. To understand learning of parameters for probabilistic graphical models
5. To understand actions and decisions with probabilistic graphical models

Unit	Description	Instructional Hours
I	**REPRESENTATION** Probability Theory, Graphs, Bayesian network representation: Bayes networks, Independence in graphs – Undirected graphical models: Parameterization, Markov Network independencies – Conditional Bayesian networks.	9
II	**TEMPLATE BASED REPRESENTATION** Temporal models (Dynamic Bayesian networks , Hidden Markov Models) – Directed probabilistic models for object-relational domains – Inference in temporal models: Kalman filters.	9
III	**INFERENCE** Exact inference: Variable elimination – Exact inference: Clique trees (Junction trees) – Approximate inference: Forward sampling, Importance sampling, MCMC – MAP inference: Variable elimination for MAP, Max-product in clique trees.	9
IV	**LEARNING** Learning graphical models – Parameter estimation: maximum-likelihood estimation, MLE for Bayesian networks, Bayesian parameter estimation – Structure learning in Bayesian networks: Constraint based, structure scores, structure search – Partially observed data: Parameter estimation, Learning models with hidden variables – Learning undirected models: Maximum likelihood	9
V	**ACTIONS AND DECISIONS** Causality – Utilities and decisions – Structured decision problems	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Construct Bayesian networks
CO2: Reason with Bayesian networks and with Dynamic networks and Hidden Markov Models
CO3: Conduct inferences with Bayesian networks
CO4: Implement algorithms to learn probabilistic graphical models
CO5: Explain actions and decisions with probabilistic graphical models

REFERENCE BOOKS :

R1- Daphne Koller and Nir Friedman, "Probabilistic Graphical Models: Principles and Techniques", MIT Press, 2009.
R2- David Barber, "Bayesian Reasoning and Machine Learning", Cambridge University Press, 2012.
R3- Adnan Darwiche, "Modeling and Reasoning with Bayesian networks", Cambridge University Press, 2009.
R4- Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
R5- Stuart Russel and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third Edition, Prentice Hall, 2009.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP1305	IMAGE PROCESSING AND ANALYSIS	3	0	0	3

Course Objective

1. To understand the basics of digital images and noise models
2. To understand spatial domain filters and frequency domain filters
3. To learn basic image analysis --- segmentation, edge detection, and corner detection
4. To learn morphological operations and texture analysis
5. To understand processing of color images and image compression techniques

Unit	Description	Instructional Hours
I	**SPATIAL DOMAIN PROCESSING** Introduction to image processing – imaging modalities – image file formats – image sensing and acquisition – image sampling and quantization – noise models – spatial filtering operations – histograms – smoothing filters – sharpening filters – fuzzy techniques for spatial filtering – spatial filters for noise removal	9
II	**FREQUENCY DOMAIN PROCESSING** Frequency domain – Review of Fourier Transform (FT), Discrete Fourier Transform (DFT), and Fast Fourier Transform (FFT) – filtering in frequency domain – image smoothing – image sharpening – selective filtering – frequency domain noise filters – wavelets – Haar Transform – multiresolution expansions – wavelet transforms – wavelets based image processing	9
III	**SEGMENTATION AND EDGE DETECTION** Thresholding techniques – region growing methods – region splitting and merging – adaptive thresholding – threshold selection – global valley – histogram concavity – edge detection – template matching – gradient operators – circular operators – differential edge operators – hysteresis thresholding – Canny operator – Laplacian operator – active contours – object segmentation	9
IV	**INTEREST POINTS, MORPHOLOGY, AND TEXTURE** Corner and interest point detection – template matching – second order derivatives – median filter based detection – Harris interest point operator – corner orientation – local invariant feature detectors and descriptors – morphology – dilation and erosion – morphological operators – grayscale morphology – noise and morphology – texture – texture analysis – co-occurrence matrices – Laws' texture energy approach – Ade's eigen filter approach	9
V	**COLOR IMAGES AND IMAGE COMPRESSION** Color models – pseudo colors – full-color image processing – color transformations – smoothing and sharpening of color images – image segmentation based on color – noise in color images. Image Compression – redundancy in images – coding redundancy – irrelevant information in images – image compression models – basic compression methods – digital image watermarking.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Explain image modalities, sensing, acquisition, sampling, and quantization, image noise models
CO2: Implement spatial filter operations and frequency domain filters
CO3: Explain frequency domain transformations
CO4: Apply segmentation algorithms, edge detection techniques, corner ,interest point detection algorithms and morphological operations
CO5: Perform texture analysis, analyze color images and Implement image compression algorithms

REFERENCE BOOKS :

R1- E. R. Davies, "Computer & Machine Vision", Fourth Edition, Academic Press, 2012.
R2- W. Burger and M. Burge, "Digital Image Processing: An Algorithmic Introduction using Java", Springer, 2008.
R3- John C. Russ, "The Image Processing Handbook", Sixth Edition, CRC Press, 2011.

R4- R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Third Edition, Pearson,2008.
R5- Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision”, Third Edition, Academic Press, 2012.
R6- D. L. Baggio et al., “Mastering OpenCV with Practical Computer Vision Projects”, Packt Publishing, 2012.
R7- Jan Erik Solem, “Programming Computer Vision with Python: Tools and algorithms for analyzing images”, O'Reilly Media, 2012

ELECTIVE II

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1306	RANDOMIZED ALGORITHMS	3	0	0	3

Course Objective

1. To understand the mathematical foundations needed for understanding and designing randomized algorithms
2. To appreciate the need for randomized algorithms
3. To expose the students to probabilistic methods
4. To understand the concept of random walk
5. To expose the students to different types of applications of randomized algorithmsGain knowledge about Microscopic techniques

Unit	Description	Instructional Hours
I	**INTRODUCTION TO RANDOMIZED ALGORITHMS** Introduction to Randomized Algorithms - Min-cut – Elementary Probability Theory – Models of Randomized Algorithms – Classification of Randomized Algorithms – Paradigms of the Design of Randomized Algorithms - Game Theoretic Techniques – Game Tree Evaluation – Minimax Principle – Randomness and Non Uniformity.	9
II	**PROBABILISTIC METHODS** Moments and Deviations – occupancy Problems – Markov and Chebyshev Inequalities – Randomized Selection – Two Point Sampling – The Stable Marriage Problem – The Probabilistic Method – Maximum Satisfiability – Expanding Graphs – Method of Conditional Probabilities – Markov Chains and Random Walks – 2-SAT Example – Random Walks on Graphs – Random Connectivity.	9
III	**ALGEBRAIC TECHNIQUES AND APPLICATIONS** Fingerprinting Techniques – Verifying Polynomial Identities – Perfect Matching in Graphs – Pattern Matching – Verification of Matrix Multiplication - Data Structuring Problems – Random Treaps – Skip Lists – Hash Tables.	9
IV	**GEOMETRIC AND GRAPH ALGORITHMS** Randomized Incremental Construction – Convex Hulls – Duality – Trapezoidal Decompositions – Linear Programming – Graph Algorithms – Min-cut – Minimum Spanning Trees.	9
V	**HASHING AND ONLINE ALGORITHMS** Hashing – Universal Hashing - Online Algorithms – Randomized Online Algorithms – Online Paging – Adversary Models – Relating the Adversaries – The k-server Problem.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the need for randomized algorithms
CO2: Discuss the classification of randomized algorithms
CO3: Present the various paradigms for designing randomized algorithms
CO4: Discuss the different probabilistic methods used for designing randomized algorithms
CO5: Apply the techniques studied to design algorithms for different applications like matrix multiplication, hashing, linear programming

REFERENCE BOOKS :

R1- Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms", Cambridge University Press, 1995.
R2- Juraj Hromkovic,"Design and Analysis of Randomized Algorithms", Springer, 2010.
R3- Michael Mitzenmacher and Eli Upfal, "Probabilty and Computing – Randomized Algorithms and Probabilistic Analysis", Cambridge University Press, 2005.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1307	MOBILE AND PERVASIVE COMPUTING	3	0	0	3

Course Objective

1. To understand the basics of Mobile Computing and Personal Computing
2. To learn the role of cellular networks in Mobile and Pervasive Computing
3. To expose to the concept of sensor and mesh networks
4. To expose to the context aware and wearable computing
5. To learn to develop applications in mobile and pervasive computing environment

Unit	Description	Instructional Hours
I	**INTRODUCTION** Differences between Mobile Communication and Mobile Computing – Contexts and Names – Functions – Applications and Services – New Applications – Making Legacy Applications Mobile Enabled – Design Considerations – Integration of Wireless and Wired Networks – Standards Bodies – Pervasive Computing – Basics and Vision – Principles of Pervasive Computing – Categories of Pervasive Devices	9
II	**3G AND 4G CELLULAR NETWORKS** Migration to 3G Networks – IMT 2000 and UMTS – UMTS Architecture – User Equipment – Radio Network Subsystem – UTRAN – Node B – RNC functions – USIM – Protocol Stack – CS and PS Domains – IMS Architecture – Handover – 3.5G and 3.9G a brief discussion – 4G LAN and Cellular Networks – LTE – Control Plane – NAS and RRC – User Plane – PDCP, RLC and MAC – WiMax IEEE 802.16d/e – WiMax Internetworking with 3GPP	9
III	**SENSOR AND MESH NETWORKS** Sensor Networks – Role in Pervasive Computing – In Network Processing and Data Dissemination – Sensor Databases – Data Management in Wireless Mobile Environments – Wireless Mesh Networks – Architecture – Mesh Routers – Mesh Clients – Routing – Cross Layer Approach – Security Aspects of Various Layers in WMN – Applications of Sensor and Mesh networks	9
IV	**CONTEXT AWARE COMPUTING & WEARABLE COMPUTING** Adaptability – Mechanisms for Adaptation - Functionality and Data – Transcoding – Location Aware Computing – Location Representation – Localization Techniques – Triangulation and Scene Analysis – Delaunay Triangulation and Voronoi graphs – Types of Context – Role of Mobile Middleware – Adaptation and Agents – Service Discovery Middleware Health BAN- Medical and Technological Requirements-Wearable Sensors-Intra-BAN communications	9
V	**APPLICATION DEVELOPMENT** Three tier architecture - Model View Controller Architecture - Memory Management – Information Access Devices – PDAs and Smart Phones – Smart Cards and Embedded Controls – J2ME – Programming for CLDC – GUI in MIDP – Application Development ON Android and iPhone	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Design a basic architecture for a pervasive computing environment
CO2: Design and allocate the resources on the 3G-4G wireless networks
CO3: Analyze the role of sensors in Wireless networks and work out the routing in mesh network
CO4: Deploy the location and context information for application development
CO5: Develop mobile computing applications based on the paradigm of context aware computing and wearable computing

REFERENCE BOOKS :

R1- Asoke K Talukder, Hasan Ahmed, Roopa R Yavagal, "Mobile Computing: Technology, Applications and Service Creation", 2nd ed, Tata McGraw Hill, 2010.

R2- Reto Meier, “Professional Android 2 Application Development”, Wrox Wiley,2010.

R3-Pei Zheng and Lionel M Li, ‘Smart Phone & Next Generation Mobile Computing’, Morgan Kaufmann Publishers, 2006.

R4- Frank Adelstein, ‘Fundamentals of Mobile and Pervasive Computing’, TMH, 2005

R5- Jochen Burthardt et al, ‘Pervasive Computing: Technology and Architecture of Mobile Internet Applications’, Pearson Education, 2003

R6- Feng Zhao and Leonidas Guibas, ‘Wireless Sensor Networks’, Morgan Kaufmann Publishers, 2004

R7- Uwe Hansmaan et al, ‘Principles of Mobile Computing’, Springer, 2003

R8- Reto Meier, “Professional Android 2 Application Development”, Wrox Wiley,2010.

R9- Mohammad s. Obaidat et al, “Pervasive Computing and Networking”,John wiley

R10- Stefan Poslad, “Ubiquitous Computing: Smart Devices, Environments and Interactions”, Wiley, 2009

R11- Frank Adelstein Sandeep K. S. Gupta Golden G. Richard III Loren Schwiebert “Fundamentals of Mobile and Pervasive Computing, “, McGraw-Hill, 2005

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1308	PARALLEL PROGRAMMING PARADIGMS	3	0	0	3

Course Objective

1. To understand models of and issues in concurrency in computing
2. To develop message-passing parallel programs using MPI
3. To develop shared-memory parallel programs using Pthreads
4. To develop shared-memory parallel programs using OpenMP
5. To use GPU for parallel programming using OpenCL and CUDA

Unit	Description	Instructional Hours
I	**FOUNDATIONS OF PARALLEL PROGRAMMING** Motivation for parallel programming - Concurrency in computing – basics of processes, multiprocessing, and threads – cache – cache mappings – caches and programs – virtual memory – instruction level parallelism – hardware multi-threading – SIMD – MIMD – interconnection networks – cache coherence – shared-memory model – issues in shared-memory model – distributed-memory model – issues in distributed-memory model – hybrid model – I/O – performance of parallel programs – parallel program design	9
II	**MESSAGE PASSING PARADIGM** Basic MPI programming – MPI_Init and MPI_Finalize – MPI communicators – SPMD programs – message passing – MPI_Send and MPI_Recv – message matching – MPI I/O – parallel I/O – collective communication – MPI_Reduce – MPI_Allreduce – broadcast – scatter – gather – allgather – derived types – remote memory access – dynamic process management – MPI for grids – performance evaluation of MPI programs	9
III	**SHARED MEMORY PARADIGM: PTHREADS** Basics of Pthreads – thread synchronization – critical sections – busy-waiting – mutexes – semaphores – barriers and condition variables – read-write locks – Caches, cache coherence and false sharing – thread safety – Pthreads case study	9
IV	**SHARED MEMORY PARADIGM: OPENMP** Basic OpenMP constructs – scope of variabls – reduction clause – parallel for directive – loops in OpenMP – scheduling loops – synchronization in OpenMP – Case Study: Producer-Consumer problem – cache issues – threads safety in OpenMP – OpenMP best practices	9
V	**GRAPHICAL PROCESSING PARADIGMS: OPENCL AND CUDA** Introduction to CUDA – CUDA programming examples – CUDA execution model – CUDA memory hierarchy – CUDA case study - introduction to OpenCL – OpenCL programming examples – Programs and Kernels – Buffers and Images – Event model – OpenCL case study	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Explain models of parallel programming and hardware level support for concurrency
CO2: Explain issues in parallel programming
CO3: Develop message-passing parallel programs using MPI framework
CO4: Develop shared-memory parallel programs using Pthreads and using OpenMP
CO5: Develop CUDA programs and OpenCL programs

REFERENCE BOOKS :

R1- Peter S. Pacheco, “An introduction to parallel programming”, Morgan Kaufmann, 2011.

R2- M. J. Quinn, “Parallel programming in C with MPI and OpenMP”, Tata McGraw Hill, 2003.

R3- W. Gropp, E. Lusk, and R. Thakur, “Using MPI-2: Advanced features of the message passing interface”,

MIT Press, 1999.

R4- W. Gropp, E. Lusk, and A. Skjellum, “Using MPI: Portable parallel programming with the message passing interface”, Second Edition, MIT Press, 1999.

R5- B. Chapman, G. Jost, and Ruud van der Pas, “Using OpenMP”, MIT Press, 2008.

R6- D. R. Butenhof, “Programming with POSIX Threads”, Addison Wesley, 1997.

R7- B. Lewis and D. J. Berg, “Multithreaded programming with Pthreads”, Sun Microsystems Press, 1998.

R8- A. Munshi, B. Gaster, T. G. Mattson, J. Fung, and D. Ginsburg, “OpenCL programming guide”, Addison Wesley, 2011.

R9- Rob Farber, “CUDA application design and development”, Morgan Haufmann, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1309	SOFTWARE REQUIREMENTS ENGINEERING	3	0	0	3

Course Objective

1. Understand system requirements
2. Identify different types of requirement
3. Generate requirements be elicitation
4. Develop requirements documentation
5. Evaluate the requirements

Unit	Description	Instructional Hours
I	**DOMAIN UNDERSTANDING** Introduction – Types of requirements – Requirements engineering process – Validating requirements – Requirements and design – Requirements and test cases – introduction to business domain – Problem analysis – Fish bone diagram – Business requirements – Business process modeling – Business use cases – Business modeling notations – UML Activity diagrams.	9
II	**REQUIREMENTS ELICITATION** Introduction – Understanding stakeholders' needs – Elicitation techniques – interviews, questionnaire, workshop, brainstorming, prototyping – Documenting stakeholders' needs	9
III	**FUNCTIONAL REQUIREMENTS** Introduction – Features and Use cases – Use case scenarios – Documenting use cases – Levels of details – SRS documents.	9
IV	**QUALITY ATTRIBUTES AND USER EXPERIENCE** Quality of solution – Quality attributes – Eliciting quality attributes – Quality attribute workshop (QAW) – Documenting quality attributes – Six part scenarios – Usability requirements – Eliciting and documenting usability requirements – Modeling user experience – Specifying UI design	9
V	**MANAGING REQUIREMENTS** Defining scope of the project – Context diagram – Managing requirements – Requirements properties – Traceability – Managing changes – Requirements metrics – Requirements management tools.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Define a process for requirements engineering and execute a process for gathering requirements through elicitation techniques.
CO2: Validate requirements according to criteria such as feasibility, clarity, preciseness etc.
CO3: Develop and document functional requirements for different types of systems.
CO4: Develop and document quality attributes of the system to be implemented and communicate the requirements to stakeholders
CO5: Negotiate with stakeholders in order to agree on a set of requirements and detect and resolve feature interactions

REFERENCE BOOKS :

R1- Axel van Lamsweerde, "Requirements Engineering", Wiley, 2009
R2- Gerald Kotonya, Ian Sommerville, "Requirements Engineering: Processes and Techniques", John Wiley and Sons, 1998
R3- Dean Leffingwell and Don Widrig, “Managing Software Requirements: A Use Case Approach (2nd Edition) ”, Addison-wesley, 2003
R4- SEI Report, “Quality Attributes Workshop”, http://www.sei.cmu.edu/library/abstracts/reports/03tr016.cfm , 2003
R5- J Nielsen, “Usability Engineering”, Academic Press, 1993

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP1310	SPEECH PROCESSING AND SYNTHESIS	3	0	0	3

Course Objective

1. To understand the mathematical foundations needed for speech processing
2. To understand the basic concepts and algorithms of speech processing and synthesis
3. To familiarize the students with the various speech signal representation, coding and recognition techniques
4. To appreciate the use of speech processing in current technologies
5. To expose the students to real– world applications of speech processing

Unit	Description	Instructional Hours
I	**FUNDAMENTALS OF SPEECH PROCESSING** Introduction – Spoken Language Structure – Phonetics and Phonology – Syllables and Words – Syntax and Semantics – Probability, Statistics and Information Theory – Probability Theory – Estimation Theory – Significance Testing – Information Theory.	9
II	**SPEECH SIGNAL REPRESENTATIONS AND CODING** Overview of Digital Signal Processing – Speech Signal Representations – Short time Fourier Analysis – Acoustic Model of Speech Production – Linear Predictive Coding – Cepstral Processing – Formant Frequencies – The Role of Pitch – Speech Coding – LPC Coder.	9
III	**SPEECH RECOGNITION** Hidden Markov Models – Definition – Continuous and Discontinuous HMMs – Practical Issues – Limitations. Acoustic Modeling – Variability in the Speech Signal – Extracting Features – Phonetic Modeling – Adaptive Techniques – Confidence Measures – Other Techniques.	9
IV	**TEXT ANALYSIS** Lexicon – Document Structure Detection – Text Normalization – Linguistic Analysis – Homograph Disambiguation – Morphological Analysis – Letter-to-sound Conversion – Prosody – Generation schematic – Speaking Style – Symbolic Prosody – Duration Assignment – Pitch Generation	9
V	**SPEECH SYNTHESIS** Attributes – Formant Speech Synthesis – Concatenative Speech Synthesis – Prosodic Modification of Speech – Source-filter Models for Prosody Modification – Evaluation of TTS Systems.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the various temporal, spectral and cepstral features required for identifying speech units – phoneme, syllable and word
CO2: Determine and apply Mel-frequency cepstral coefficients for processing all types of signals
CO3: Justify the use of formant and concatenative approaches to speech synthesis
CO4: Identify the apt approach of speech synthesis depending on the language to be processed
CO5: Determine the various encoding techniques for representing speech.

REFERENCE BOOKS :

R1- Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, "Spoken Language Processing – A guide to Theory, Algorithm and System Development", Prentice Hall PTR, 2001.
R2- Thomas F.Quatieri, "Discrete-Time Speech Signal Processing", Pearson Education, 2002.
R3- Lawrence Rabiner and Biing-Hwang Juang, "Fundamentals of Speech Recognition", Prentice Hall Signal Processing Series, 1993.
R4- Sadaoki Furui, "Digital Speech Processing: Synthesis, and Recognition, Second Edition, (Signal Processing and Communications)", Marcel Dekker, 2000.
R5- Joseph Mariani, "Language and Speech Processing", Wiley, 2009.

ELECTIVE III

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2301	MACHINE LEARNING TECHNIQUES	3	0	0	3

Course Objective

1. To understand the machine learning theory
2. To implement linear and non-linear learning models
3. To implement distance-based clustering techniques
4. To build tree and rule based models
5. To apply reinforcement learning techniques

Unit	Description	Instructional Hours
I	**FOUNDATIONS OF LEARNING** Components of learning – learning models – geometric models – probabilistic models – logic models – grouping and grading – learning versus design – types of learning – supervised – unsupervised – reinforcement – theory of learning – feasibility of learning – error and noise – training versus testing – theory of generalization – generalization bound – approximation-generalization tradeoff – bias and variance – learning curve	9
II	**CLASSIFICATION AND LINEAR MODELS** Classification-General approach-Decision tree induction-Rule based classification- Linear models-linear classification-univariate linear regression-multivariate linear regression-regularized regression-logistic regression-perceptrons-Advanced classification methods-multilayer feed forward neural networks-learning neural networks structures-support vector machines-Classification using Frequent patterns-other classification methods.	9
III	**DISTANCE-BASED MODELS** Nearest neighbor models – K-means – clustering around medoids – silhouttes – hierarchical clustering – Density based methods- Grid based methods- Advanced cluster analysis- k-d trees – locality sensitive hashing – non-parametric regression – bagging and random forests – boosting – meta learning	9
IV	**TREE AND RULE MODELS** Decision trees – learning decision trees – ranking and probability estimation trees – regression trees – clustering trees – learning ordered rule lists – learning unordered rule lists – descriptive rule learning – Mining Frequent patterns, Association and Correlations, advanced association rule techniques-first order rule learning	9
V	**REINFORCEMENT LEARNING** Passive reinforcement learning – direct utility estimation – adaptive dynamic programming – temporal-difference learning – active reinforcement learning – exploration – learning an action utility function – Generalization in reinforcement learning – policy search – applications in game playing – applications in robot control	9
	Total Instructional Hours	**45**

Course Outcome

CO1: To explain theory underlying machine learning
CO2: To construct algorithms to learn linear and non-linear models
CO3: To implement data clustering algorithms
CO4: To construct algorithms to learn tree and rule-based models
CO5: To apply reinforcement learning techniques

REFERENCE BOOKS :

R1 - Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, “Learning from Data”, AMLBook Publishers,

2012.
R2 - P. Flach, “Machine Learning: The art and science of algorithms that make sense of data”, ambridge University Press, 2012.
R3 - K. P. Murphy, “Machine Learning: A probabilistic perspective”, MIT Press, 2012.
R4 - C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2007.
R5 - D. Barber, “Bayesian Reasoning and Machine Learning”, Cambridge University Press, 2012.
R6 - M. Mohri, A. Rostamizadeh, and A. Talwalkar, “Foundations of Machine Learning”, MIT Press, 2012.
R7 - T. M. Mitchell, “Machine Learning”, McGraw Hill, 1997.
R8 - S. Russel and P. Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, Prentice Hall, 2009.
R9 - Jiawei Han, Micheline Kamber , Jian Pei, “Data Mining: Concepts and Techniques”, Third Edition (The Morgan Kaufmann Series in Data Management Systems), 2012.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2302	REAL TIME SYSTEMS	3	0	0	3

Course Objective

1. To provide good understanding of fundamental concepts in real time systems.
2. To provide understanding of advanced topics in real time systems.
3. To provide understanding on basic multi-task scheduling algorithms for periodic, aperiodic, and sporadic tasks as well as understand the impact of the latter two on scheduling
4. To expose to understand capabilities of commercial off-the-shelf R-T kernel.
5. To expose to real time communications and databases

Unit	Description	Instructional Hours
I	**INTRODUCTION** Real-time systems – Applications – Basic Model – Characteristics – Safety and Reliability – Real-Time tasks – Timing Constraints – Modelling Timing Constraints.	9
II	**SCHEDULING REAL-TIME TASKS** Concepts – Types of RT Tasks and their Characteristics – Task Scheduling – Clock-Driven Scheduling – Hybrid Schedulers - Event-Driven Scheduling – EDF Scheduling – RMA – Issues with RMA – Issues in Using RMA in Practical Situations	9
III	**RESOURCE SHARING AMONG RT TASKS & SCHEDULING RT TASKS** Resource Sharing Among RT Tasks – Priority Inversion – PIP – HLP – PCP – Types of Priority Inversions Under PCP – Features of PCP – Issues in using Resource Sharing Protocol – Handling Task Dependencies – Multiprocessor Task Allocation – Dynamic Allocation of Tasks – Fault-Tolerant Scheduling of Tasks – Clocks in Distributed RT Systems – Centralized and Distributed Clock Synchronization.	9
IV	**COMMERCIAL RT OPERATING SYSTEMS** Time Services – Features of RT OS – Unix as a RT OS – Unix Based RT OS – Windows as a RT OS – POSIX – Survey of RTOS: PSOS – VRTX – VxWorks – QNX - μC/OS-II – RT Linux – Lynx – Windows CE – Benching RT Systems.	9
V	**RT COMMUNICATION & DATABASES** Examples of Applications Requiring RT Communication – Basic Concepts – RT Communication in a LAN – Soft & Hard RT Communication in a LAN – Bounded Access Protocols for LANs – Performance Comparison – RT Communication Over Packet Switched Networks – QoS Framework – Routing – Resource Reservation – Rate Control – QoS Models – Examples Applications of RT Databases – RT Databases – Characteristics of Temporal Data – Concurrency Control in RT Databases – Commercial RT Databases.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Understand the basics and importance of real-time systems
CO2: Generate a high-level analysis and design document based on requirements specifications
CO3: Generate a test plan based on requirements specification and a validation plan based on all documentation
CO4: Understand basic multi-task scheduling algorithms for periodic, aperiodic, and sporadic tasks as well as understand the impact of the latter two on scheduling
CO5: Understand capabilities of at least one commercial off-the-shelf R-T kernel

REFERENCE BOOKS :

R1 - Rajib Mall, "Real-Time Systems: Theory and Practice," Pearson, 2008.
R2 - Jane W. Liu, "Real-Time Systems" Pearson Education, 2001.
R3 - Krishna and Shin, "Real-Time Systems," Tata McGraw Hill. 1999.
R4 - Alan C. Shaw, "Real-Time Systems and Software", Wiley, 2001.
R5 - Philip Laplante, "Real-Time Systems Design and Analysis", 2nd Edition, Prentice Hall of India.
R6 - Resource Management in Real-time Systems and Networks, C. Siva Ram Murthy and G. Manimaran, MIT Press, March 2001.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2303	COMPUTER VISION	3	0	0	3

Course Objective

1. To review image processing techniques for computer vision
2. To understand shape and region analysis
3. To understand Hough Transform and its applications to detect lines, circles, ellipses
4. To understand three-dimensional image analysis techniques and motion analysis
5. To study some applications of computer vision algorithms

Unit	Description	Instructional Hours
I	**IMAGE PROCESSING FOUNDATIONS** Review of image processing techniques – classical filtering operations – thresholding techniques – edge detection techniques – corner and interest point detection – mathematical morphology – texture	9
II	**SHAPES AND REGIONS** Binary shape analysis – connectedness – object labeling and counting – size filtering – distance functions – skeletons and thinning – deformable shape analysis – boundary tracking procedures – active contours – shape models and shape recognition – centroidal profiles – handling occlusion – boundary length measures – boundary descriptors – chain codes – Fourier descriptors – region descriptors – moments	9
III	**HOUGH TRANSFORM** Line detection – Hough Transform (HT) for line detection – foot-of-normal method – line localization – line fitting – RANSAC for straight line detection – HT based circular object detection – accurate center location – speed problem – ellipse detection – Case study: Human Iris location – hole detection – generalized Hough Transform (GHT) – spatial matched filtering – GHT for ellipse detection – object location – GHT for feature collation	9
IV	**3D VISION AND MOTION** Methods for 3D vision – projection schemes – shape from shading – photometric stereo – shape from texture – shape from focus – active range finding – surface representations – point-based representation – volumetric representations – 3D object recognition – 3D reconstruction – introduction to motion – triangulation – bundle adjustment – translational alignment – parametric motion – spline-based motion – optical flow – layered motion	9
V	**APPLICATIONS** Application: Photo album – Face detection – Face recognition – Eigen faces – Active appearance and 3D shape models of faces Application: Surveillance – foreground-background separation – particle filters – Chamfer matching, tracking, and occlusion – combining views from multiple cameras – human gait analysis Application: In-vehicle vision system: locating roadway – road markings – identifying road signs – locating pedestrians	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Implement fundamental image processing techniques required for computer vision
CO2: Perform shape analysis and implement boundary tracking techniques
CO3: Apply chain codes and other region descriptors
CO4: Apply Hough Transform for line, circle, and ellipse detections and 3D vision techniques
CO5: Implement motion related techniques and develop applications using computer vision techniques

REFERENCE BOOKS :

R1 - E. R. Davies, "Computer & Machine Vision", Fourth Edition, Academic Press, 2012.
R2 - R. Szeliski, "Computer Vision: Algorithms and Applications", Springer 2011.
R3 - Simon J. D. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012.
R4 - Mark Nixon and Alberto S. Aquado, "Feature Extraction & Image Processing for Computer Vision", Third Edition, Academic Press, 2012.

R5 - D. L. Baggio et al., “Mastering OpenCV with Practical Computer Vision Projects”, PacktPublishing, 2012.
R6 - Jan Erik Solem, “Programming Computer Vision with Python: Tools and algorithms for analyzing images”, O'Reilly Media, 2012.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2304	NETWORK AND INFORMATION SECURITY	3	0	0	3

Course Objective

1. To understand the fundamentals of Cryptography
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

Unit	Description	Instructional Hours
I	**INTRODUCTION** An Overview of Computer Security-Security Services-Security Mechanisms-Security Attacks-Access Control Matrix, Policy-Security policies, Confidentiality policies, Integrity policies and Hybrid policies	9
II	**CRYPTOSYSTEMS & AUTHENTICATION** Classical Cryptography-Substitution Ciphers-permutation Ciphers-Block Ciphers-DES- Modes of Operation- AES-Linear Cryptanalysis, Differential Cryptanalysis- Hash Function - SHA 512-Message Authentication Codes-HMAC - Authentication Protocols	9
III	**PUBLIC KEY CRYPTOSYSTEMS** Introduction to Public key Cryptography- Number theory- The RSA Cryptosystem and Factoring Integer- Attacks on RSA-The ELGamal Cryptosystem- Digital Signature Algorithm-Finite Fields-Elliptic Curves Cryptography- Key management – Session and Interchange keys, Key exchange and generation-PKI	9
IV	**SYSTEM IMPLEMENTATION** Design Principles, Representing Identity, Access Control Mechanisms, Information Flow and Confinement Problem Secure Software Development: Secured Coding - OWASP/SANS Top Vulnerabilities – Buffer Overflows - Incomplete mediation - XSS - Anti Cross Site Scripting Libraries - Canonical Data Format - Command Injection - Redirection - Inference – Application Controls	9
V	**NETWORK SECURITY** Secret Sharing Schemes-Kerberos- Pretty Good Privacy (PGP)-Secure Socket Layer (SSL)-Intruders – HIDS- NIDS - Firewalls – Viruses	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Implement basic security algorithms required by any computing system.
CO2: Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
CO3: Analyze the possible security attacks in complex real time systems and their effective countermeasures
CO4: Identify the security issues in the network and resolve it and evaluate security mechanisms using rigorous approaches, including theoretical derivation, modeling, and simulations
CO5: Formulate research problems in the computer security field

REFERENCE BOOKS :

R1 - William Stallings, "Cryptography and Network Security: Principles and Practices", Third Edition, Pearson Education, 2006.
R2 - Matt Bishop ,"Computer Security art and science ", Second Edition, Pearson Education, 2002
R3 - Wade Trappe and Lawrence C. Washington, "Introduction to Cryptography with Coding Theory" Second

Edition, Pearson Education, 2007
R4 - Jonathan Katz, and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press, 2007
R5 - Douglas R. Stinson, "Cryptography Theory and Practice", Third Edition, Chapman & Hall/CRC, 2006
R6 - Wenbo Mao, "Modern Cryptography – Theory and Practice", Pearson Education, First Edition, 2006.
R7 - Network Security and Cryptography, Menezes Bernard, Cengage Learning, New Delhi, 2011
R8 - Man Young Rhee, Internet Security, Wiley, 2003
R9 - OWASP top ten security vulnerabilities: http://xml.coverpages.org/OWASP-TopTen.pdf

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2305	SOFTWARE ARCHITECTURES	3	0	0	3

Course Objective

1. Understand architectural requirements
2. Identify architectural structures
3. Develop architectural documentation
4. Generate architectural alternatives
5. Evaluate the architecture against the drivers

Unit	Description	Instructional Hours
I	**ARCHITECTURAL DRIVERS** Introduction – Standard Definitions of Software Architecture– Architectural structures – Influence of software architecture on organization – Architecture Business Cycle – Functional requirements – Technical constraints – Quality Attributes – Quality Attribute Workshop (QAW) – Documenting Quality Attributes – Six part scenarios	9
II	**ARCHITECTURAL VIEWS AND DOCUMENTATION** Introduction – Standard Definitions for views – Structures and views- Perspectives: Static, dynamic and physical and the accompanying views – Representing views-available notations – Good practices in documentation– Documenting the Views using UML – Merits and Demerits of using visual languages – Need for formal languages - Architectural Description Languages – ACME	9
III	**ARCHITECTURAL STYLES** Introduction – Data flow styles – Call-return styles – Shared Information styles – Event styles – Case studies for each style	9
IV	**ARCHITECTURAL DESIGN** Approaches for architectural design – System decomposition – Attributes driven design – Architecting for specific quality attributes – Performance, Availability – Security – Architectural conformance	9
V	**ARCHITECTURE EVALUATION AND SOME SPECIAL TOPICS** Need for evaluation – Scenario based evaluation against the drivers – ATAM and its variations – Case studies in architectural evaluations – SOA and Web services – Cloud Computing – Adaptive structures	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Explain key architectural drivers and the influence of architecture on business and technical activities
CO2: Identify key architectural structures
CO3: Adopt good practices for documenting the architecture
CO4: Develop alternative architectures for a given problem and explain how to use formal languages to specify architecture
CO5: Evaluate the architecture against the drivers and describe the recent trends in software architecture

REFERENCE BOOKS :

R1- Len Bass, Paul Clements, and Rick Kazman, “Software Architectures Principles and Practices”, 2n Edition, \ Addison-Wesley, 2003.
R2 - Anthony J Lattanze, “Architecting Software Intensive System. A Practitioner's Guide”, Auerbach Publications, 2010.
R3 - Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, and Judith Stafford, “Documenting Software Architectures. Views and Beyond”, 2nd Edition, Addison-Wesley, 2010.
R4 - Paul Clements, Rick Kazman, and Mark Klein, “Evaluating software architectures: Methods and case studies.”, Addison-Wesley, 2001.

R5 - David Garlan and Mary Shaw, "Software architecture: Perspectives on an emerging discipline",Prentice Hall, 1996.

R6 - Rajkumar Buyya, James Broberg, and Andrzej Goscinski, "Cloud Computing. Principles and Paradigms", John Wiley & Sons, 2011

R7 - Mark Hansen, "SOA Using Java Web Services", Prentice Hall, 2007

R8 - David Garlan, Bradley Schmerl, and Shang-Wen Cheng, "Software Architecture-Based Self- Adaptation," 31-56. Mieso K Denko, Laurence Tianruo Yang, and Yan Zang (eds.), "Autonomic Computing and Networking". Springer Verlag, 2009.

ELECTIVE IV

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2306	BIO-INSPIRED COMPUTING	3	0	0	3

Course Objective

1. Learn evolutionary theory and algorithms
2. Understand Cellular Automata and artificial life
3. Learn artificial neural systems and related learning algorithms
4. Understand behavioral systems especially in the context of Robotics
5. To know about the components and building blocks hypothesis of Genetic Algorithm and understand collective systems such as ACO, PSO, and swarm robotics

Unit	Description	Instructional Hours
I	**EVOLUTIONARY AND CELLULAR SYSTEMS** Foundations of evolutionary theory – Genotype – artificial evolution – genetic representations – initial population – fitness functions – selection and reproduction – evolutionary measures – evolutionary algorithms – evolutionary electronics – evolutionary algorithm case study Cellular systems – cellular automata – modeling with cellular systems – computation with cellular systems – artificial life – analysis and synthesis of cellular systems	9
II	**NEURAL SYSTEMS** Biological nervous systems – artificial neural networks – neuron models – architecture – signal encoding – synaptic plasticity – unsupervised learning – supervised learning – reinforcement learning – evolution of neural networks – Dynamic Programming-Neuro dynamics - hybrid neural systems – case study	9
III	**BEHAVIORAL SYSTEMS** Behavior is cognitive science – behavior in AI – behavior based robotics – biological inspiration for robots – robots as biological models – robot learning – evolution of behavioral systems – learning in behavioral systems – co-evolution of body and control – towards self reproduction – simulation and reality	9
IV	**GENETIC ALGORITHM** Introduction (GA) – application of GA - building blocks hypothesis-Representation-Fitness measures-Genetic operators-GA based machine learning.	9
V	**COLLECTIVE SYSTEMS** Biological self-organization – Particle Swarm Optimization (PSO) – ant colony optimization (ACO) – swarm robotics – co-evolutionary dynamics – artificial evolution of competing systems – artificial evolution of cooperation – case study	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Implement and apply evolutionary algorithms
CO2: Explain cellular automata and artificial life
CO3: Implement and apply neural systems and explain behavioral systems
CO4:Discover knowledge to develop Genetic algorithm
CO5: Implement and apply collective intelligence systems

REFERENCE BOOKS :

R1 - D. Floreano and C. Mattiussi, "Bio-Inspired Artificial Intelligence", MIT Press, 2008.
R2 - F. Neumann and C. Witt, “Bioinspired Computation in combinatorial optimization: Algorithms and their computational complexity”, Springer, 2010.
R3 - A. E. Elben and J. E. Smith, “Introduction to Evolutionary Computing”, Springer, 2010.
R4 - D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley, 1989.

R5 - Simon O. Haykin, “Neural Networks and Learning Machines”, Third Edition, Prentice Hall, 2008.
R6 - M. Dorigo and T. Stutzle, “Ant Colony Optimization”, A Bradford Book, 2004.
R7 - R. C. Ebelhart et al., “Swarm Intelligence”, Morgan Kaufmann, 2001.8.
R8 - S.N.Sivanandam, S.N.Deepa, “Introduction to Genetic Algorithms”, Springer, 2008 Edition.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2307	CLOUD COMPUTING	3	0	0	3

Course Objective

1. To introduce the broad perceptive of cloud architecture and model
2. To understand the concept of Virtualization and be familiar with the lead players in cloud.
3. To understand the features of cloud simulator and to apply different cloud programming model as per need.
4. To be able to set up a private cloud and to understand the design of cloud Services.
5. To learn to design the trusted cloud Computing system

Unit	Description	Instructional Hours
I	**CLOUD ARCHITECTURE AND MODEL** Technologies for Network-Based System – System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture. Cloud Models:- Characteristics – Cloud Services – Cloud models (IaaS, PaaS, SaaS) – Public vs Private Cloud –Cloud Solutions - Cloud ecosystem – Service management – Computing on demand.	9
II	**VIRTUALIZATION** Basics of Virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Virtual Clusters and Resource management – Virtualization for Data-center Automation.	9
III	**CLOUD INFRASTRUCTURE** Architectural Design of Compute and Storage Clouds – Layered Cloud Architecture Development – Design Challenges - Inter Cloud Resource Management – Resource Provisioning and Platform Deployment – Global Exchange of Cloud Resources.	9
IV	**PROGRAMMING MODEL** Parallel and Distributed Programming Paradigms – MapReduce , Twister and Iterative MapReduce – Hadoop Library from Apache – Mapping Applications - Programming Support - Google App Engine, Amazon AWS - Cloud Software Environments -Eucalyptus, Open Nebula, OpenStack, Aneka, CloudSim	9
V	**SECURITY IN THE CLOUD** Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security.	9
	Total Instructional Hours	**45**

Course Outcome

CO1:Compare the strengths and limitations of cloud computing
CO2: Identify the architecture, infrastructure and delivery models of cloud computing
CO3: Apply suitable virtualization concept.
CO4: Choose the appropriate cloud player and the appropriate Programming Models and approach.
CO5: Address the core issues of cloud computing such as security, privacy and interoperability, design Cloud Services and set a private cloud

REFERENCE BOOKS :

R1 - Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
R2 - John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.

R3 - Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", TMH, 2009.
R4 - Kumar Saurabh, " Cloud Computing – insights into New-Era Infrastructure", Wiley India,2011.
R5 - George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud" O'Reilly
R6 - James E. Smith, Ravi Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
R7 - Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, "Grid and Cloud Computing – A Business Perspective on Technology and Applications", Springer.
R8 - Ronald L. Krutz, Russell Dean Vines, "Cloud Security – A comprehensive Guide to Secure Cloud Computing", Wiley – India, 2010.
R9 - Rajkumar Buyya, Christian Vecchiola, S.Tamarai Selvi, 'Mastering Cloud Computing", TMGH,2013.
R10 - Gautam Shroff, Enterprise Cloud Computing, Cambridge University Press, 2011
R11 - Michael Miller, Cloud Computing, Que Publishing,2008
R12 - Nick Antonopoulos, Cloud computing, Springer Publications, 2010

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP2308	DATA VISUALIZATION TECHNIQUES	3	0	0	3

Course Objective

1. To introduce visual perception and core skills for visual analysis
2. To understand visualization for time-series analysis and ranking analysis
3. To understand visualization for deviation analysis and distribution analysis
4. To understand visualization for correlation analysis and multivariate analysis
5. To understand issues and best practices in information dashboard design

Unit	Description	Instructional Hours
I	**CORE SKILLS FOR VISUAL ANALYSIS** Information visualization – effective data analysis – traits of meaningful data – visual perception – making abstract data visible – building blocks of information visualization – analytical interaction – analytical navigation – optimal quantitative scales – reference lines and regions – trellises and crosstabs – multiple concurrent views – focus and context – details on demand – over-plotting reduction – analytical patterns – pattern examples	9
II	**TIME-SERIES, RANKING, AND DEVIATION ANALYSIS** Time-series analysis – time-series patterns – time-series displays – time-series best practices – part-to-whole and ranking patterns – part-to-whole and ranking displays – best practices – deviation analysis – deviation analysis displays – deviation analysis best practices	9
III	**DISTRIBUTION, CORRELATION, AND MULTIVARIATE ANALYSIS** Distribution analysis – describing distributions – distribution patterns – distribution displays – distribution analysis best practices – correlation analysis – describing correlations – correlation patterns – correlation displays – correlation analysis techniques and best practices – multivariate analysis – multivariate patterns – multivariate displays – multivariate analysis techniques and best practices	9
IV	**INFORMATION DASHBOARD DESIGN I** Information dashboard – categorizing dashboards – typical dashboard data – dashboard design issues and best practices – visual perception – limits of short-term memory – visually encoding data – Gestalt principles – principles of visual perception for dashboard design	9
V	**INFORMATION DASHBOARD DESIGN II** Characteristics of dashboards – key goals in visual design process – dashboard display media – designing dashboards for usability – meaningful organization – maintaining consistency – aesthetics of dashboards – testing for usability – case studies: sales dashboard, CIO dashboard, Telesales dashboard, marketing analysis dashboard	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Explain principles of visual perception
CO2: Apply core skills for visual analysis
CO3: Apply visualization techniques for various data analysis tasks
CO4: Design information dashboard
CO5: Design CIO dashboard and telesales dashboard

REFERENCE BOOKS :

R1 - Stephen Few, "Now you see it: Simple Visualization techniques for quantitative analysis", Analytics Press, 2009.

R2 - Stephen Few, "Information dashboard design: The effective visual communication of data", O'Reilly, 2006.

R3 - Edward R. Tufte, "The visual display of quantitative information", Second Edition, Graphics Press, 2001.

R4 - Nathan Yau, "Data Points: Visualization that means something", Wiley, 2013.

R5 - Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", O'Reilly, 2008.

R6 - Gert H. N. Laursen and Jesper Thorlund, "Business Analytics for Managers: Taking business intelligence beyond reporting", Wiley, 2010.

R7 - Evan Stubbs, "The value of business analytics: Identifying the path to profitability", Wiley, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2309	PROTOCOLS AND ARCHITECTURE FOR WIRELESS SENSOR NETWORKS	3	0	0	3

Course Objective

1. To introduce the basis of wireless sensor networks
2. To understand the concept of WSN architecture
3. To understand the WSN deployment and configuration.
4. To be able to set up routing protocol for data manipulation
5. To learn to basis of WSN platform and tools

Unit	Description	Instructional Hours
I	**INTRODUCTION AND OVERVIEW OF WIRELESS SENSOR NETWORKS** Background of Sensor Network Technology, Application of Sensor Networks, Challenges for Wireless Sensor Networks, Mobile Adhoc NETworks (MANETs) and Wireless Sensor Networks, Enabling Technologies For Wireless Sensor Networks.	9
II	**ARCHITECTURES** Single-node Architecture, Hardware Components & Design Constraints, Operating Systems and Execution Environments, Introduction to TinyOS and nesC, Network Architecture, Sensor Network Scenarios, Optimization Goals and Figures of Merit, Design Principles for WSNs, Service Interfaces of WSNs, Gateway Concepts.	9
III	**DEPLOYMENT AND CONFIGURATION** Localization and Positioning, Coverage and Connectivity, Single-hop and Multi-hop Localization, Self Configuring Localization Systems, Sensor Management Network Protocols: Issues in Designing MAC Protocol for WSNs, Classification of MAC Protocols, S-MAC Protocol, B-MAC Protocol, IEEE 802.15.4 Standard and Zig Bee, Dissemination Protocol for Large Sensor Network.	9
IV	**ROUTING PROTOCOLS AND DATA MANIPULATION** Issues in Designing Routing Protocols, Classification of Routing Protocols, Energy-Efficient Routing, Unicast, Broadcast and Multicast, Geographic Routing. Data Centric and Content based Routing, Storage and Retrieval in Network, Compression Technologies for WSN, Data Aggregation Technique.	9
V	**SENSOR NETWORK PLATFORMS AND TOOLS** Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level Software Platforms, Node-level Simulators, State-centric Programming.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Compare the strengths and limitations of WSN.
CO2: Identify the architecture
CO3: Apply suitable configuration
CO4: Choose the appropriate routing
CO5: Address the core issues WSN tools and platform.

REFERENCE BOOKS :

R1 - Holger Karl & Andreas Willig, “Protocols And Architectures for Wireless Sensor Networks", John Wiley, 2005.

R2 - Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

R3 - Raghavendra, Cauligi S, Sivalingam, Krishna M., Zanti Taieb, “Wireless Sensor Network”, Springer 1st Ed. 2004 (ISBN: 978-4020-7883-5).

R4 - Kazem Sohraby, Daniel Minoli, & Taieb Znati, “Wireless Sensor Networks- Technology, Protocols, and Applications”, John Wiley, 2007.

R5 - N. P. Mahalik, “Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications” Springer Verlag.

R6 - Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
	16CP2310	LANGUAGE TECHNOLOGIES	3	0	0	3

Course Objective

1. To understand the mathematical foundations needed for language processing and processing of Morphology and Part-of Speech Taggers
2. To understand different aspects of natural language syntax and the various methods used for processing syntax
3. To understand different methods of disambiguating word senses and various applications of natural language processing
4. To learn the indexing and searching processes of a typical information retrieval system and to study NLP based retrieval systems
5. To gain knowledge about typical text categorization and clustering techniques

Unit	Description	Instructional Hours
I	**INTRODUCTION** Natural Language Processing – Mathematical Foundations – Elementary Probability Theory – Essential information Theory - Linguistics Essentials - Parts of Speech and Morphology – Phrase Structure – Semantics – Corpus Based Work.	9
II	**WORDS** Collocations – Statistical Inference – n-gram Models – Word Sense Disambiguation – Lexical Acquisition.	9
III	**GRAMMAR** Markov Models – Part-of-Speech Tagging – Probabilistic Context Free Grammars - Parsing.	9
IV	**INFORMATION RETRIEVAL** Information Retrieval Architecture – Indexing - Storage – Compression Techniques – Retrieval Approaches – Evaluation - Search Engines - Commercial Search Engine Features – Comparison - Performance Measures – Document Processing - NLP based Information Retrieval – Information Extraction.	9
V	**TEXT MINING** Categorization – Extraction Based Categorization – Clustering - Hierarchical Clustering - Document Classification and Routing - Finding and Organizing Answers from Text Search – Text Categorization and Efficient Summarization using Lexical Chains – Machine Translation – Transfer Metaphor - Interlingual and Statistical Approaches.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the different linguistic components of given sentences
CO2: Design a morphological analyser for a language of your choice using finite state automata concepts
CO3: Implement a parser by providing suitable grammar and words
CO4: Discuss algorithms for word sense disambiguation and build a tagger to semantically tag words using WordNet
CO5: Design an application that uses different aspects of language processing.

REFERENCE BOOKS :

R1 - Christopher D.Manning and Hinrich Schutze, " Foundations of Statistical Natural Language Processing ", MIT Press, 1999.
R2 - Daniel Jurafsky and James H. Martin, " Speech and Language Processing" , Pearson, 2008.
R3 - Ron Cole, J.Mariani, et.al "Survey of the State of the Art in Human Language Technology", Cambridge University Press, 1997.
R4 - Michael W. Berry, " Survey of Text Mining: Clustering, Classification and Retrieval", Springer Verlag, 2003.

ELECTIVE V

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3301	SOCIAL NETWORK ANALYSIS	3	0	0	3

Course Objective

1. To understand the components of the social network
2. To model and visualize the social network
3. To mine the users in the social network
4. To understand the evolution of the social network
5. To mine the interest of the user

Unit	Description	Instructional Hours
I	**INTRODUCTION** Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Statistical Properties of Social Networks -Network analysis - Development of Social Network Analysis - Key concepts and measures in network analysis - Discussion networks - Blogs and online communities - Web-based networks	9
II	**MODELING AND VISUALIZATION** Visualizing Online Social Networks - A Taxonomy of Visualizations - Graph Representation - Centrality- Clustering - Node-Edge Diagrams - Visualizing Social Networks with Matrix-Based Representations- Node-Link Diagrams - Hybrid Representations - Modelling and aggregating social network data – Random Walks and their Applications –Use of Hadoop and Map Reduce - Ontological representation of social individuals and relationships.	9
III	**MINING COMMUNITIES** Aggregating and reasoning with social network data, Advanced Representations – Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Evaluating Communities – Core Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Node Classification in Social Networks.	9
IV	**EVOLUTION** Evolution in Social Networks – Framework - Tracing Smoothly Evolving Communities - Models and Algorithms for Social Influence Analysis - Influence Related Statistics - Social Similarity and Influence - Influence Maximization in Viral Marketing - Algorithms and Systems for Expert Location in Social Networks - Expert Location without Graph Constraints - with Score Propagation – Expert Team Formation - Link Prediction in Social Networks - Feature based Link Prediction – Bayesian Probabilistic Models - Probabilistic Relational Models	9
V	**TEXT AND OPINION MINING** Text Mining in Social Networks -Opinion extraction – Sentiment classification and clustering - Temporal sentiment analysis - Irony detection in opinion mining - Wish analysis - Product review mining – Review Classification – Tracking sentiments towards topics over time	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Work on the internals components of the social network
CO2: Model and visualize the social network
CO3: Mine the behaviour of the users in the social network
CO4: Predict the possible next outcome of the social network
CO5: Mine the opinion of the user

REFERENCE BOOKS :

R1 - Charu C. Aggarwal, "Social Network Data Analytics", Springer; 2011
R2 - Peter Mika, "Social Networks and the Semantic Web", Springer, 1st edition, 2007.
R3 - Borko Furht, "Handbook of Social Network Technologies and Applications", Springer, 1st edition, 2010.

R4 - Guandong Xu , Yanchun Zhang and Lin Li, "Web Mining and Social Networking – Techniques and applications", Springer, 1st edition, 2011.

R5 - Giles, Mark Smith, John Yen, "Advances in Social Network Mining and Analysis", Springer, 2010.

R6 - Ajith Abraham, Aboul Ella Hassanien, Vaclav Snašel, "Computational Social Network Analysis: Trends, Tools and Research Advances", Springer, 2009.

R7 - Toby Segaran, "Programming Collective Intelligence", O'Reilly, 2012.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3302	MANAGING BIG DATA	3	0	0	3

Course Objective

1. Learn business case studies for big data analytics
2. Understand nosql big data management
3. Perform map-reduce analytics using Hadoop
4. Understand the mapreduce functionalities
5. Apply Hadoop using tools for various applications

Unit	Description	Instructional Hours
I	**UNDERSTANDING BIG DATA** What is big data – why big data – convergence of key trends – unstructured data – industry examples of big data – web analytics – big data and marketing – fraud and big data – risk and big data – credit risk management – big data and algorithmic trading – big data and healthcare – big data in medicine – advertising and big data – big data technologies – introduction to Hadoop – open source technologies – cloud and big data – mobile business intelligence – Crowd sourcing analytics – inter and trans firewall analytics	9
II	**NOSQL DATA MANAGEMENT** Introduction to NoSQL – aggregate data models – aggregates – key-value and document data models – relationships – graph databases – schemaless databases – materialized views – distribution models – sharding – master-slave replication – peer-peer replication – sharding and replication – consistency – relaxing consistency – version stamps – map-reduce – partitioning and combining – composing map-reduce calculations	9
III	**BASICS OF HADOOP** Data format – analyzing data with Hadoop – scaling out – Hadoop streaming – Hadoop pipes – design of Hadoop distributed file system (HDFS) – HDFS concepts – Java interface – data flow – Hadoop I/O – data integrity – compression – serialization – Avro – file-based data structures	9
IV	**MAPREDUCE APPLICATIONS** MapReduce workflows – unit tests with MRUnit – test data and local tests – anatomy of MapReduce job run – classic Map-reduce – YARN – failures in classic Map-reduce and YARN – job scheduling – shuffle and sort – task execution – MapReduce types – input formats – output formats	9
V	**HADOOP RELATED TOOLS** Hbase – data model and implementations – Hbase clients – Hbase examples – praxis.Cassandra – cassandra data model – cassandra 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.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Describe big data and use cases from selected business domains
CO2:Explain NoSQL big data management
CO3: Install, configure, and run Hadoop and HDFS
CO4: Perform map-reduce analytics using Hadoop
CO5: Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics

REFERENCE BOOKS :

R1 - Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
R2 - P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
R3 - Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
R4 - Eric Sammer, "Hadoop Operations", O'Reilley, 2012.
R5 - E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
R6 - Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
R7 - Eben Hewitt, "Cassandra: The Definitive Guide", O'Reilley, 2010.
R8 - Alan Gates, "Programming Pig", O'Reilley, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3303	MODEL CHECKING AND PROGRAM VERIFICATION	3	0	0	3

Course Objective

1. To understand automata for model checking
2. To understand LTL, CTL, and CTL*
3. To understand timed automata, TCTL, and PCTL
4. To understand verification of deterministic and recursive programs
5. To understand verification of object-oriented programs and to understand verification of parallel, distributed, and non-deterministic programs

Unit	Description	Instructional Hours
I	**AUTOMATA AND TEMPORAL LOGICS** Automata on finite words – model checking regular properties – automata on infinite words – Buchi automata – Linear Temporal Logic (LTL) – automata based LTL model checking – Computational Tree Logic (CTL) – CTL model checking – CTL* model checking	9
II	**TIMED AND PROBABILISTIC TREE LOGICS** Timed automata – timed computational tree logic (TCTL) – TCTL model checking – probabilistic systems – probabilistic computational tree logic (PCTL) – PCTL model checking – PCTL* - Markov decision processes	9
III	**VERIFYING DETERMINISTIC AND RECURSIVE PROGRAMS** Introduction to program verification – verification of "while" programs – partial and total correctness – verification of recursive programs – case study: binary search – verifying recursive programs with parameters	9
IV	**VERIFYING OBJECT-ORIENTED AND PARALLEL PROGRAMS** Partial and total correctness of object-oriented programs – case study: Insertion in linked lists – verification of disjoint parallel programs – verifying programs with shared variables – case study: parallel zero search – verification of synchronization – case study: the mutual exclusion problem	9
V	**VERIFYING NON-DETERMINISTIC AND DISTRIBUTED PROGRAMS** Introduction to non-deterministic programs – partial and total correctness of non-deterministic programs – case study: The Welfare Crook Problem – syntax and semantics of distributed programs – verification of distributed programs – case study: A Transmission Problem – introduction to fairness	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Perform model checking using LTL and Perform model checking using CTL
CO2: Perform model checking using CTL*
CO3: Perform model checking using TCTL and PCTL
CO4: Verify deterministic and recursive programs
CO5: Verify object-oriented programs AND Verify parallel, distributed, and non-deterministic programs

REFERENCE BOOKS :

R1 - C. Baier, J.-P. Katoen, and K. G. Larsen, "Principles of Model Checking", MIT Press, 2008.
R2 - E. M. Clarke, O. Grumberg, and D. A. Peled, "Model Checking", MIT Press, 1999.
R3 - M. Ben-Ari, "Principles of the SPIN Model Checker", Springer, 2008.
R4 - K. R. Apt, F. S. de Boer, E.-R. Olderog, and A. Pnueli, "Verification of Sequential and Concurrent Programs", Third Edition, Springer, 2010.
R5 - M. Huth and M. Ryan, "Logic in Computer Science --- Modeling and Reasoning about Systems", Second Edition, Cambridge University Press, 2004.
R6 - B. Berard et al., "Systems and Software Verification: Model-checking techniques and tools", Springer, 2010.
R7 - J. B. Almeida, M. J. Frade, J. S. Pinto, and S. M. de Sousa, "Rigorous Software Development: An Introduction to Program Verification", Springer, 2011.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3304	MEDICAL IMAGE PROCESSING	3	0	0	3

Course Objective

1. To Understand the basics of medical image processing
2. To Understand the Storage Methods.
3. Discuss about the visualization
4. To understand the classification
5. To study about resolution

Unit	Description	Instructional Hours
I	**INTRODUCTION** Introduction to medical imaging technology, systems, and modalities. Brief history; importance; applications; trends; challenges. Medical Image Formation Principles: X-Ray physics; X-Ray generation, attenuation, scattering; dose Basic principles of CT; reconstruction methods; artifacts; CT hardware.	9
II	**STORAGE AND PROCESSING** Medical Image Storage, Archiving and Communication Systems and Formats Picture archiving and communication system (PACS); Formats: DICOM Radiology Information Systems (RIS) and Hospital Information Systems (HIS). Medical Image Processing, Enhancement, Filtering Basic image processing algorithms Thresholding; contrast enhancement; SNR characteristics; filtering; histogram modeling.	9
III	**VISUALIZATION** Medical Image Visualization Fundamentals of visualization; surface and volume rendering/visualization; animation; interaction. Magnetic Resonance Imaging (MRI) Mathematics of MR; spin physics; NMR spectroscopy; imaging principles and hardware; image artifacts.	9
IV	**SEGMENTATION AND CLASSIFICATION** Medical Image Segmentation - Histogram-based methods; Region growing and watersheds; Markov Random Field models; active contours; model-based segmentation. Multi-scale segmentation; semi-automated methods; clustering-based methods; classification-based methods; atlas-guided approaches; multi-model segmentation. Medical Image Registration Intensity-based methods; cost functions; optimization techniques.	9
V	**NUCLEAR IMAGING** PET and SPECT Ultrasound Imaging methods; mathematical principles; resolution; noise effect; 3D imaging; positron emission tomography; single photon emission tomography; ultrasound imaging; applications. Medical Image Search and Retrieval Current technology in medical image search, content-based image retrieval, new trends: ontologies. Applications. Other Applications of Medical Imaging Validation, Image Guided Surgery, Image Guided Therapy, Computer Aided Diagnosis/Diagnostic Support Systems.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Apply Medical Image Formation Principles
CO2: Apply medical image storage and hospital information system
CO3:Develop visualization for medical images
CO4: Design segmentation and classification techniques
CO5: Apply nuclear imagimg for PET

REFERENCE BOOKS :

R1 - Paul Suetens, "Fundamentals of Medical Imaging", Second Edition, Cambridge University Press, 2009.
R2 - J. Michael Fitzpatrick and Milan Sonka, "Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis", SPIE Publications, 2009.
R3 - Kayvan Najarian and Robert Splinter, "Biomedical Signal and Image Processing", Second Edition, CRC Press, 2005.
R4 - Geoff Dougherty, "Digital Image Processing for Medical Applications", First Edition, Cambridge University Press, 2009.

R5 - Jerry L. Prince and Jonathan Links, "Medical Imaging Signals and Systems", First Edition, Prentice Hall, 2005.
R6 - John L. Semmlow, "Biosignal and Medical Image Processing", Second Edition, CRC Press, 2008.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3305	SOFTWARE DESIGN	3	0	0	3

Course Objective

1. Analyze specifications
2. Describe approaches to design
3. Develop design documentation
4. Evaluate the design
5. Understand user centered design

Unit	Description	Instructional Hours
	SOFTWARE DESIGN PRINCIPLES	
I	Introduction – Design process – Managing complexity – Software modeling and notations – Abstraction – Modularity – Hierarchy – Coupling - Cohesion – Design guidelines and checklists – Refactoring	9
	OO DESIGN	
II	Object model – Classes and objects – Object oriented analysis – Key abstractions and mechanisms – Object oriented design – Identifying design elements – Detailed design – Case studies.	9
	DESIGN PATTERNS	
III	Introduction to patterns – Design context – Reusable solutions – Documenting reusable solutions – Standard patterns from GOF book.	9
	FUNCTION AND SERVICE ORIENTED DESIGNS	
IV	Structural decomposition – Detailed Design – Function oriented design Case study – Services – Service identification – Service design – Service composition – choreography and orchestration – Service oriented design Case study	9
	USER CENTERED DESIGN AND DESIGN REVIEW	
V	Introduction to user centered design – Use in context – Interface and interaction – User centered design principles – Task analysis – Evaluation – Introduction to design review– Testing the design – Walk throughs – Review against check lists.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Describe different approaches to designing a software application
CO2: Analyze specifications and identify appropriate design strategies.
CO3: Develop an appropriate design for a given set of requirements
CO4:Identify applicable design patterns for the solution
CO5:Evaluate a given design against the specifications

REFERENCE BOOKS :

R1 - Grady Booch et al., "Object Oriented Analysis and Design with Applications", 3rd Edition, Pearson, 2010.
R2 - Carlos Otero, "Software Engineering Design: Theory and Practice", CRC Press, 2012
R3 - David Budgen, "Software Design", 2nd Edtion, Addison Wesley, 2003
R4 - Alan Shalloway and James R Trott, "Design Patterns Explained: A New Perspective on Object-Oriented Design", 2nd Edition, Addison-Wesley Professional, 2004
R5 - Hassan Gomaa, "Software Modeling and Design", Cambridge University Press, 2011
R6 - Eric Gamma et al., "Design Patterns: Elements of Reusable Object-Oriented Software",Addison-Wesley Professional, 1994
R7 - Ian Sommerville, "Software Engineering", 9th Edition, Addison-Wesley, 2010
R8 - M B Rosson and J M Carroll, "Usability Engineering: Scenario-Based Development of Human-Computer Interaction", Morgan Kaufmann, 2002

ELECTIVE VI

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3306	MULTI OBJECTIVE OPTIMIZATION TECHNIQUES	3	0	0	3

Course Objective

1. Learn fundamental principles of Multiobjective Optimization (MOP)
2. Survey different Multiobjective Optimization algorithms
3. Introduce various design issues of MOP
4. Develop and Evaluate MOP Algorithms
5. Learn Parallel and hybrid MOP Algorithms and learn other Metaheuristics

Unit	Description	Instructional Hours
I	**INTRODUCTION AND CLASSICAL APPROACHES** Multiobjective optimization: Introduction - Multiobjective optimization problem-principles – Difference between single and multiobjective optimization – Dominance and Pareto Optimality , Classical Methods – Weighted Sum - □ Constraint method – Weighted Metric methods – Benson's method - Value Function - Goal Programming methods – Interactive Methods	9
II	**MOP EVOLUTIONARY ALGORITHMS** Generic MOEA - Various MOEAs: MOGA, NSGA-II, NPGA, PAES, SPEA2, MOMGA, micro GA - Constrained MOEAs: Penalty Function approach - Constrained Tournament – Ray – Tai –Seow's Method.	9
III	**THEORETICAL ISSUES** Fitness Landscapes - Fitness Functions - Pareto Ranking - Pareto Niching and Fitness Sharing - Recombination Operators - Mating Restriction - Solution Stability and Robustness - MOEA Complexity - MOEA Scalability - Running Time Analysis - MOEA Computational Cost - No Free Lunch Theorem.	9
IV	**MOEA TESTING, ANALYSIS, AND PARALLELIZATION** MOEA Experimental Measurements – MOEA Statistical Testing Approaches – MOEA Test Suites - MOEA Parallelization: Background – Paradigms – Issues - MOEA Local Search Techniques.	9
V	**APPLICATIONS AND ALTERNATIVE METAHEURISTICS** Scientific Applications: Computer Science and Computer Engineering - Alternative Metaheuristics: Simulated Annealing – Tabu Search and Scatter Search – Ant System – Distributed Reinforcement Learning – Particle Swarm Optimization – Differential Evolution – Artificial Immune Systems - Other Heuristics.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Explain MOP principles
CO2:Explain classical methods to solve MOP problems
CO3: Be familiar with and explain structures of different MOP algorithms
CO4: Solve constrained MOP problems
CO5: Perform a evaluation and analysis of MOP algorithm results

REFERENCE BOOKS :

R1 - Carlos A. Coello Coello, Gary B. Lamont, David A. Van Veldhuizen, "Evolutionary Algorithms for Solving Multi-objective Problems", Second Edition, Springer, 2007.

R2 - Kalyanmoy Deb, " Multi-Objective Optimization Using Evolutionary Algorithms", John Wiley, 2002.

R3 - Aimin Zhoua, Bo-Yang Qub, Hui Li c, Shi-Zheng Zhaob, Ponnuthurai Nagaratnam Suganthan b, Qingfu Zhangd, "Multiobjective evolutionary algorithms: A survey of the state of the art", Swarm and volutionary Computation (2011) 32–49.

R4 - E Alba, M Tomassini, "Parallel and evolutionary algorithms", Evolutionary Computation, IEEE ransactions on 6 (5), 443-462.

R5 - Crina Grosan, Ajith Abraham, "Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews", Studies in Computational Intelligence, Vol. 75, Springer, 2007.

R6 - Christian Blum and Andrea Roli. 2003. Metaheuristics in combinatorial optimization:Overview and conceptual comparison. ACM Comput. Surv. 35, 3 (September 2003), 268-308.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3307	INFORMATION STORAGE MANAGEMENT	3	0	0	3

Course Objective

1. Understand basics of storage technology
2. Analyze storage system architecture for host environment
3. Understand networked storage and its architectures
4. Analyze monitoring and managing datacenter
5. Evaluate information availability and storage virtualization

Unit	Description	Instructional Hours
I	**INTRODUCTION TO STORAGE TECHNOLOGY** Review data creation and the amount of data being created and understand the value of data to a business, challenges in data storage and data management, Solutions available for data storage, Core elements of a data center infrastructure, role of each element in supporting business activities	9
II	**STORAGE SYSTEMS ARCHITECTURE** Hardware and software components of the host environment, Key protocols and concepts used by each component ,Physical and logical components of a connectivity environment ,Major physical components of a disk drive and their function, logical constructs of a physical disk, access characteristics, and performance Implications, Concept of RAID and its components, Different RAID levels and their suitability for different application environments: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5, RAID 0+1, RAID 1+0, RAID 6, Compare and contrast integrated and modular storage systems ,Iligh-level architecture and working of an intelligent storage system	9
III	**INTRODUCTION TO NETWORKED STORAGE** Evolution of networked storage, Architecture, components, and topologies of FC-SAN, NAS, and IP-SAN, Benefits of the different networked storage options, understand the need for long-term archiving solutions and describe how CAS full fill the need, understand the appropriateness of the different networked storage options for different application environments	9
IV	**INFORMATION AVAILABILITY, MONITORING & MANAGING DATACENTER** List reasons for planned/unplanned outages and the impact of downtime, Impact of downtime - Differentiate between business continuity (BC) and disaster recovery (DR) ,RTO and RPO, Identify single points of failure in a storage infrastructure and list solutions to mitigate these failures, Architecture of backup/recovery and the different backup/ recovery topologies, replication technologies and their role in ensuring information availability and business continuity, Remote replication technologies and their role in providing disaster recovery and business continuity capabilities. Identify key areas to monitor in a data center, Industry standards for data center monitoring and management, Key metrics to monitor for different components in a storage infrastructure, Key management tasks in a data center	9
V	**SECURING STORAGE AND STORAGE VIRTUALIZATION** Information security, Critical security attributes for information systems, Storage security domains, List and analyzes the common threats in each domain, Virtualization technologies, block-level and file-level virtualization technologies and processes	9
	Total Instructional Hours	**45**

Course Outcome

CO1:Explain data creation and its applications
CO2: Apply storage system architecture and key protocols for host environment
CO3: Apply FC-SAN and NAS
CO4: Identify information availability and monitoring and managing data center
CO5: Describe storage virtualization and security in storage

REFERENCE BOOKS :

R1 - EMC Corporation, Information Storage and Management, Wiley, India.
R2 - Robert Spalding, "Storage Networks: The Complete Reference", Tata McGraw Hill , Osborne, 2003.
R3 - Marc Farley, "Building Storage Networks", Tata McGraw Hill ,Osborne, 2001.
R4 - Additional resource material on www.emc.com/resource-library/resource-library.esp

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3308	SOFTWARE QUALITY ASSURANCE	3	0	0	3

Course Objective

1. Describe approaches to quality assurance
2. Understand quality models
3. Analyze the system based on the chosen quality model
4. Understand structural testing and its adequacy criteria
5. Understand functional testing and design test cases

Unit	Description	Instructional Hours
I	**INTRODUCTION** Introduction – Views on quality – Cost of quality - Quality models – Quality frameworks – Verification and Validation – Defect taxonomy – Defect management – Statistics and measurements – IEEE standards – Quality assurance and control processes	9
II	**VERIFICATION** Introduction – Verification techniques – Inspections, reviews, walk-throughs – Case studies	9
III	**TEST GENERATION** Software testing- Validation – Test plan – Test cases - Test Generation – Equivalence partitioning – Boundary value analysis – Category partition method – Combinatorial generation – Decision tables – Examples and Case studies	9
IV	**STRUCTURAL TESTING** Introduction – Test adequacy criteria – Control flow graph – Coverages: block, conditions, multiple conditions, MC/DC, path – Data flow graph – Definition and use coverages – C-use, P-use, Defclear, Def-use – Finite state machines – Transition coverage – Fault based testing – Mutation analysis – Case studies	9
V	**FUNCTIONAL TESTING** Introduction – Test adequacy criteria - Test cases from use cases – Exploratory testing - Integration, system, acceptance, regression testing – Testing for specific attributes: Performance, load and stress testing – Usability testing – Security testing - Test automation – Test oracles	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Describe different approaches to testing software applications
CO2: Analyze specifications and identify appropriate test generation strategies
CO3: Develop an appropriate test design for a given test object
CO4: Identify applicatble measurements for the verification and validation effort
CO5:Execute the test design and Evaluate the testing effort based on adequate measures

REFERENCE BOOKS :

R1 - Boriz Beizer, "Software Testing Techniques", 2nd Edition, DreamTech, 2009.
R2 - Aditya P. Mathur, "Foundations of Software Testing", Pearson, 2008
R3 - Mauro Pezze and Michal Young, "Software Testing and Analysis. Process, Principles, and Techniques", John Wiley 2008
R4 - Stephen H. Kan, "Metrics and Models in Software Quality Engineering", 2nd Edition, Pearson, 2003
R5 - Kshirasagar Naik and Priyadarshi Tripathy (Eds), "Software Testing and Quality Assurance: Theory and Practice", John Wiley, 2008
R6 - "Combinatorial Methods in Software Testing", http://csrc.nist.gov/groups/SNS/acts/index.html

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3309	GREEN COMPUTING	3	0	0	3

Course Objective

1. To introduce the concept of green computing.
2. To create awareness of energy efficient computing.
3. To understand the power management in computing devices
4. To analyze the consumption of power in data centers
5. Analyze IBM green tectnology and its applications

Unit	Description	Instructional Hours
I	**INTRODUCTION** Energy- efficient – power efficient and thermal aware computing and communication - Newton‟s cooling model and basic thermodynamics and sustainability.	9
II	**POWER MANAGEMENT** Operating system Directed power management – Power management history and motivation – key power management concepts – power management scenarios – ACPI desktop motherboard design	9
III	**DEVELOPMENT OF EFFICIENT POWER MANAGEMENT SYSTEM** Dual mode desktop power delivery – system BIOS – Designing mobile systems - Communication with peripheral devices – Drivers – Developing robust power managed applications	9
IV	**ENERGY EFFICIENT DATA CENTER** Data center power consumption – Power metrics – Energy efficient data center tuning - energy efficient server management – Industry vision and recommendations	9
V	**CASE STUDIES AND APPLICATION** Google green datacenter - IBM green technology - Microsoft – Case Studies – Applying Green IT Strategies and Applications to a Home – Hospital - Packaging Industry and Telecom Sector	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the benefits and challenges of energy efficient computing.
CO2 : Develop energy efficient computing applications.
CO3 : Apply the strategies of going Green
CO4: Develop energy efficient data center
CO5: Design home applications using green IT strategies

REFERENCE BOOKS :

R1 - Jerzy Kolinski, Ram Chary, Andrew Henroid, and Barry Press, “Building the Power-Efficient PC A Developer's Guide to ACPI Power Management”, Intel Press August 2001.
R2 - Lauri Minas, Brad Ellison, “Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers”, Intel Press, 2009.
R3 - Bhuvan Unhelkar, “Green IT Strategies and Applications-Using Environmental Intelligence”, CRC Press, June 2011.
R4 - Wu Chun Feng, “Green Computing: Large-Scale Energy Efficiency”, CRC Press INC, 2013.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CP3310	RECONFIGURABLE COMPUTING	3	0	0	3

Course Objective

1. To understand the need for reconfigurable computing
2. To expose the students to various device architectures
3. To examine the various reconfigurable computing systems
4. To understand the different types of compute models for programming reconfigurable architectures
5. To expose the students to HDL programming and familiarize with the development Environment

Unit	Description	Instructional Hours
I	**DEVICE ARCHITECTURE** General Purpose Computing Vs Reconfigurable Computing – Simple Programmable Logic Devices – Complex Programmable Logic Devices – FPGAs – Device Architecture - Case Studies.	9
II	**RECONFIGURABLE COMPUTING ARCHITECTURES AND SYSTEMS** Reconfigurable Processing Fabric Architectures – RPF Integration into Traditional Computing Systems – Reconfigurable Computing Systems – Case Studies – Reconfiguration Management.	9
III	**PROGRAMMING RECONFIGURABLE SYSTEMS** Compute Models - Programming FPGA Applications in HDL – Compiling C for Spatial Computing – Operating System Support for Reconfigurable Computing.	9
IV	**MAPPING DESIGNS TO RECONFIGURABLE PLATFORMS** The Design Flow - Technology Mapping – FPGA Placement and Routing – Configuration Bitstream Generation – Case Studies with Appropriate Tools.	9
V	**APPLICATION DEVELOPMENT WITH FPGAS** Case Studies of FPGA Applications – System on a Programmable Chip (SoPC) Designs.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Identify the need for reconfigurable architectures
CO2:Discuss the architecture of FPGAs
CO3:Point out the salient features of different reconfigurable architectures
CO4:Build basic modules using any HDL
CO5: Develop applications using any HDL and appropriate tools and design and build an SoPC for a particular application

REFERENCE BOOKS :

R1 - Maya B. Gokhale and Paul S. Graham, "Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays", Springer, 2005.
R2 - Scott Hauck and Andre Dehon (Eds.), "Reconfigurable Computing – The Theory and Practice of FPGA-Based Computation", Elsevier / Morgan Kaufmann, 2008.
R3 - Christophe Bobda, "Introduction to Reconfigurable Computing – Architectures, Algorithms and Applications", Springer, 2010.

OPEN ELECTIVE

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CPX401	MOBILE APPLICATION DEVELOPMENT	3	0	0	3

Course Objective

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

Unit	Description	Instructional Hours
I	**INTRODUCTION** 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	9
II	**BASIC DESIGN** 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.	9
III	**ADVANCED DESIGN** 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.	9
IV	**TECHNOLOGY I – ANDROID** 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.	9
V	**TECHNOLOGY II – IOS** 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.	9
	Total Instructional Hours	**45**

Course Outcome

CO1: Describe the requirements for mobile applications
CO2: Explain the challenges in mobile application design and development
CO3: Develop design for mobile applications for specific requirements
CO4: Implement the design using Android SDK
CO5:Implement the design using Objective C and iOS

REFERENCE BOOKS :

R1 - http://developer.android.com/develop/index.html
R2 - Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012
R3 - Charlie Collins, Michael Galpin and Matthias Kappler, “Android in Practice”, DreamTech,2012
R4 - James Dovey and Ash Furrow, “Beginning Objective C”, Apress, 2012
R5 - David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, “Beginning iOS 6 Development: Exploring the iOS SDK”, Apress, 2013.

PROGRAMME	COURSE CODE	NAME OF THE COURSE	L	T	P	C
ME	16CPX402	DATA MINING TECHNIQUES	3	0	0	3

Course Objective

1. Understand basics of data mining and its algorithms
2. Analyze data mining algorithms for various applications
3. Differentiate OnLine Transaction Processing and OnLine Analytical processing
4. Learn Multidimensional schemas suitable for data warehousing
5. Understand various data mining functionalities

Unit	Description	Instructional Hours
	INTRODUCTION TO DATA MINING	
I	Introduction to Data Mining – Data Mining Tasks – Components of Data Mining Algorithms – Data Mining supporting Techniques – Major Issues in Data Mining – Measurement and Data – Data Preprocessing – Data sets	9
	OVERVIEW OF DATA MINING ALGORITHMS	
II	Overview of Data Mining Algorithms – Models and Patterns – Introduction – The Reductionist viewpoint on Data Mining Algorithms – Score function for Data Mining Algorithms- Introduction – Fundamentals of Modeling – Model Structures for Prediction – Models for probability Distributions and Density functions – The Curve of Dimensionality – Models for Structured Data – Scoring Patterns – Predictive versus Descriptive score functions – Scoring Models with Different Complexities – Evaluation of Models and Patterns – Robust Methods.	9
	CLASSIFICATIONS	
III	Classifications – Basic Concepts – Decision Tree induction – Bayes Classification Methods – Rule Based Classification – Model Evaluation and Selection – Techniques to Improve Classification Accuracy – Classification: Advanced concepts – Bayesian Belief Networks- Classification by Back Propagation – Support Vector Machine – Classification using frequent patterns.	9
	CLUSTER ANALYSIS	
IV	Cluster Analysis: Basic concepts and Methods – Cluster Analysis – Partitioning methods – Hierarchical methods – Density Based Methods – Grid Based Methods – Evaluation of Clustering – Advanced Cluster Analysis: Probabilistic model based clustering – Clustering High – Dimensional Data – Clustering Graph and Network Data – Clustering with Constraints.	9
	ASSOCIATION RULE MINING AND VISUALIZATION	
V	Association Rule Mining – Introduction – Large Item sets – Basic Algorithms – Parallel and Distributed Algorithms – Comparing Approaches – Incremental Rules – Advanced Association Rule Techniques – Measuring the Quality of Rules – Visualization of Multidimensional Data – Diagrams for Multidimensional visualization – Visual Data Mining – Data Mining Applications – Case Study: WEKA.	9
	Total Instructional Hours	**45**

Course Outcome

1. Design a data mart or data warehouse for any organization
2. Develop skills to write queries using DMQL
3. Extract knowledge using data mining techniques
4. Adapt to new data mining tools.
5. Explore recent trends in data mining such as web mi ning, spatial-temporal mining

REFERENCE BOOKS :

R1 - Jiawei Han, Micheline Kamber , Jian Pei, "Data Mining: Concepts and Techniques", Third Edition (The Morgan Kaufmann Series in Data Management Systems), 2012.
R2 - David J. Hand, Heikki Mannila and Padhraic Smyth "Principles of Data Mining" (Adaptive Computation and Machine Learning), 2005
R3 - Margaret H Dunham, "Data Mining: Introductory and Advanced Topics", 2003
R4 - Soman, K. P., Diwakar Shyam and Ajay V. "Insight Into Data Mining: Theory And Practice", PHI, 2009.