SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution, Affiliated to Anna University, Chennai)
SRIPERUMBUDUR TK.- 602 117
REGULATION – 2016
M.E. COMPUTER SCIENCE AND ENGINEERING
CURRICULUM AND SYLLABUS

SEMESTER I

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	MA16187	Applied Probability and Statistics	3	1	0	4
2	CP16101	Design and Management of Computer Networks	3	0	0	3
3	CP16102	Advanced Data Structures and Algorithms	3	0	0	3
4	CP16103	Multicore Architectures	3	0	0	3
5		Elective I	3	0	0	3
6		Elective II	3	0	0	3
PRACTICALS						
7	CP16111	Advanced Data Structures Laboratory	0	0	4	2
8	CP16112	Case Study - Network Design (Team Work)	0	0	2	1
		TOTAL	**18**	**1**	**6**	**22**

SEMESTER II

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16201	Theoretical Foundations of Computer Science	3	1	0	4
2	CP16202	Advanced Databases	3	0	0	3
3	CP16203	Principles of Programming Languages	3	0	0	3
4	CP16204	Advanced Operating Systems	3	0	0	3
5		Elective III	3	0	0	3
6		Elective IV	3	0	0	3
PRACTICALS						
7	CP16211	Advanced Database Laboratory	0	0	4	2
8	CP16212	Case Study - Operating Systems Design (Team Work)	0	0	2	1
		TOTAL	**18**	**1**	**6**	**22**

SEMESTER III

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16301	Recent Trends in Computing Technologies	3	1	0	4
2		Elective V	3	0	0	3
3		Elective VI	3	0	0	3
4		Elective VII	3	0	0	3
PRACTICALS						
5	CP16311	Project Work (Phase I)	0	0	12	6
		TOTAL	**12**	**1**	**12**	**19**

SEMESTER IV

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
PRACTICALS						
1	CP16411	Project Work (Phase II)	0	0	24	12
		TOTAL	**0**	**0**	**24**	**12**

ELECTIVE I

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16001	Formal models of software systems	3	0	0	3
2	CP16002	Performance Evaluation of Computer Systems	3	0	0	3
3	CP16003	Probabilistic Reasoning Systems	3	0	0	3
4	CP16004	Data Analysis and Business Intelligence	3	0	0	3
5	CP16005	Image Processing and Analysis	3	0	0	3
6	NW16002	Sensing Techniques and Sensors	3	0	0	3

ELECTIVE II

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16006	Randomized Algorithms	3	0	0	3
2	NW16003	Mobile and Pervasive Computing	3	0	0	3
3	CP16007	Parallel Programming Paradigms	3	0	0	3
4	CP16008	Software Requirements Engineering	3	0	0	3
5	CP16009	Speech Processing and Synthesis	3	0	0	3
6	CP16010	Machine Learning Techniques	3	0	0	3

ELECTIVE III

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16011	Concurrency Models	3	0	0	3
2	CP16012	Real Time Systems	3	0	0	3
3	CP16013	Computer Vision	3	0	0	3
4	NW16202	Network and Information Security	3	0	0	3
5	CP16014	Design and Analysis of Parallel Algorithms	3	0	0	3
6	CP16015	Software Architectures	3	0	0	3

ELECTIVE IV

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16016	Model Checking and Program Verification	3	0	0	3
2	CP16017	Embedded Software Development	3	0	0	3
3	CP16018	Cloud Computing	3	0	0	3
4	CP16019	Data Visualization Techniques	3	0	0	3
5	NW16006	Protocols and Architecture for Wireless Sensor Networks	3	0	0	3
6	CP16020	Language Technologies	3	0	0	3

ELECTIVE V

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16021	Agent Based Intelligent Systems	3	0	0	3
2	CP16022	Knowledge Engineering	3	0	0	3
3	CP16023	Business Intelligence	3	0	0	3
4	CP16024	Natural Language Processing	3	0	0	3

ELECTIVE VI

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16025	Security in Computing	3	0	0	3
2	CP16026	Adhoc and Sensor Networks	3	0	0	3
3	CP16027	High Performance Computing	3	0	0	3
4	CP16028	Embedded Real Time Operating Systems	3	0	0	3
5	CP16029	Hardware and Trouble Shooting	3	0	0	3

ELECTIVE VII

SL. No	COURSE CODE	COURSE TITLE	L	T	P	C
1	CP16030	Software Testing	3	0	0	3
2	CP16031	Design Patterns	3	0	0	3
3	CP16032	Human Computer Interaction	3	0	0	3
4	CP16033	Mobile Operating Systems	3	0	0	3
5	CP16034	Research Methodology	3	0	0	3
6	CP16036	Soft Computing	3	0	0	3

MA16187 APPLIED PROBABILITY AND STATISTICS

L	T	P	C
3	1	0	4

OBJECTIVES:

- To introduce the basic concepts of one dimensional and two dimensional Random Variables.
- To provide information about Estimation theory, Correlation, Regression and Testing of hypothesis.
- To enable the students to use the concepts of multivariate normal distribution and principle components analysis.

UNIT I ONE DIMENSIONAL RANDOM VARIABLES 9+3

Random variables - Probability function – Moments – Moment generating functions and their properties – Binomial, Poisson, Geometric, Uniform, Exponential, Gamma and Normal distributions – Functions of a Random Variable.

UNIT II TWO DIMENSIONAL RANDOM VARIABLES 9+3

Joint distributions – Marginal and Conditional distributions – Functions of two dimensional random variables – Regression Curve – Correlation.

UNIT III ESTIMATION THEORY 9+3

Unbiased Estimators – Method of Moments – Maximum Likelihood Estimation - Curve fitting by Principle of least squares – Regression Lines.

UNIT IV TESTING OF HYPOTHESES 9+3

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.

UNIT V MULTIVARIATE ANALYSIS 9+3

Random Vectors and Matrices - Mean vectors and Covariance matrices - Multivariate Normal density and its properties - Principal components Population principal components – Principal components from standardized variables.

TOTAL: 45 PERIODS

OUTCOMES:

- The student will able to acquire the basic concepts of Probability and Statistical techniques for solving mathematical problems which will be useful in solving Engineering problems.

REFERENCES:

1. Jay L. Devore, “Probability and Statistics for Engineering and the Sciences”, Thomson and Duxbury, 2002.
2. Richard Johnson. ”Miller & Freund’s Probability and Statistics for Engineer”, Prentice – Hall, Seventh Edition, 2007.
3. Richard A. Johnson and Dean W. Wichern, “Applied Multivariate Statistical Analysis”, Pearson Education, Asia, Fifth Edition, 2002.
4. Gupta S.C. and Kapoor V.K.”Fundamentals of Mathematical Statistics”, Sultan and Sons, 2001.
5. Dallas E Johnson, “Applied Multivariate Methods for Data Analysis”, Thomson an Duxbury press, 1998.

CP16101 DESIGN AND MANAGEMENT OF COMPUTER NETWORKS

L	T	P	C
3	0	0	3

UNIT I INTRODUCTION TO NETWORK MANAGEMENT **9**

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.

UNIT II REQUIREMENTS ANALYSIS **9**

Requirement Analysis Process – Gathering and Listing Requirements- Developing service metrics– Characterizing behavior – Developing RMA requirements – Developing delay Requirements -Developing capacity Requirements - Developing supplemental performance Requirements –Requirements mapping – Developing the requirements specification.

UNIT III FLOW ANALYSIS **9**

Individual and Composite Flows – Critical Flows - Identifying and developing flows – Data sources and sinks – Flow models- Flow prioritization – Flow specification algorithms – Example Applications of Flow Analysis.

UNIT IV NETWORK ARCHITECTURE **9**

Architecture and design – Component Architectures – Reference Architecture – Architecture Models – System and Network Architecture – Addressing and Routing 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.

UNIT V NETWORK DESIGN **9**

Design Concepts – Design Process - Network Layout – Design Traceability – Design Metrics – Logical Network Design – Topology Design – Bridging, Switching and Routing Protocols- Physical Network Design – Selecting Technologies and Devices for Campus and Enterprise Networks –Optimizing Network Design.

TOTAL: 45 PERIODS

REFERENCES:

1. Network Analysis, Architecture, and Design By James D. McCabe, Morgan Kaufmann, Third Edition, 2007.ISBN-13: 978-0123704801.
2. Computer Networks: A Systems Approach by Larry L. Peterson, Bruce S. Davie - 2007,Elsevier Inc.
3. 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.
4. 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.
5. "Network Design and Management" – by Steven T.Karris, Orchard publications, Second edition, Copyright 2009, ISBN 978-1-934404-15-7.
6. "Network Design, Management and Technical Perspective", Teresa C. Mann-Rubinson and Kornel Terplan, CRC Press, 1999.
7. "Ethernet Networks-Design, Implementation, Operation and Management by Gilbert Held, John Wiley and sons, Fourth Edition.
8. James Kurose and Keith Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 1999.

CP16102 ADVANCED DATA STRUCTURES AND ALGORITHMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the principles of iterative and recursive algorithms.
- To learn the graph search algorithms.
- To study network flow and linear programming problems.
- To learn the hill climbing and dynamic programming design techniques.
- To develop recursive backtracking algorithms.
- To get an awareness of NP completeness and randomized algorithms.
- To learn the principles of shared and concurrent objects.
- To learn concurrent data structures.

UNIT I ITERATIVE AND RECURSIVE ALGORITHMS 9

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.

UNIT II OPTIMISATION ALGORITHMS 9

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 Back tracking Algorithm- Pruning Branches-Satisfiability.

UNIT III DYNAMIC PROGRAMMING ALGORITHMS 9

Developing a Dynamic Programming Algorithm-Subtle Points- Question for the Little Bird-Sub instances and Sub Solutions-Set of Substances-Decreasing Time and Space-Number of Solutions-Code. Reductions and NP - Completeness – Satisfiability - Proving NP-Completeness-3-Coloring-Bipartite Matching. Randomized Algorithms - Randomness to Hide Worst Cases-Optimization Problems with a Random Structure.

UNIT IV SHARED OBJECTS AND CONCURRENT OBJECTS 9

Shared Objects and Synchronization -Properties of Mutual Exclusion-The Mora l- The Producer–Consumer Problem -The Readers–Writers Problem-Realities of Parallelization-Parallel Programming- Principles- Mutual Exclusion-Time- Critical Sections--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- Formal Definitions- Progress Conditions-The Java Memory Model.

UNIT V CONCURRENT DATA STRUCTURES 9

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Design and apply iterative and recursive algorithms.
- Design and implement optimisation algorithms in specific applications.
- Design appropriate shared objects and concurrent objects for applications.
- Implement and apply concurrent linked lists, stacks, and queues.

REFERENCES:

1. Jeff Edmonds, “How to Think about Algorithms”, Cambridge University Press, 2008.
2. M. Herlihy and N. Shavit, “The Art of Multiprocessor Programming”, Morgan Kaufmann, 2008.
3. Steven S. Skiena, “The Algorithm Design Manual”, Springer, 2008.
4. Peter Brass, “Advanced Data Structures”, Cambridge University Press, 2008.
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, “Algorithms” , McGraw Hill, 2008.
6. J. Kleinberg and E. Tardos, "Algorithm Design“, Pearson Education, 2006.
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms“, PHIL earning Private Limited, 2012.
8. Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
9. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, “The Design and Analysis of Computer Algorithms**”,** Addison-Wesley, 1975.
10. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, ”Data Structures and Algorithms”, Pearson,2006.

CP16103 MULTICORE ARCHITECTURES L T P C
3 0 0 3

OBJECTIVES:

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

UNIT I FUNDAMENTALS OF QUANTITATIVE DESIGN AND ANALYSIS 9

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.

UNIT II DLP IN VECTOR, SIMD AND GPU ARCHITECTURES 9

Vector Architecture - SIMD Instruction Set Extensions for Multimedia – Graphics Processing Units- Detecting and Enhancing Loop Level Parallelism - Case Studies.

UNIT III TLP AND MULTIPROCESSORS 9

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.

UNIT IV RLP AND DLP IN WAREHOUSE-SCALE ARCHITECTURES 9

Programming Models and Workloads for Warehouse-Scale Computers – Architectures for Warehouse-Scale Computing – Physical Infrastructure and Costs – Cloud Computing – Case Studies.

UNIT V ARCHITECTURES FOR EMBEDDED SYSTEMS 9

Embedded Systems – Signal Processing and Embedded Applications – The Digital Signal Processor – Embedded Multiprocessors - Case Studies.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

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

REFERENCES:

1. John L. Hennessey and David A. Patterson, “Computer Architecture – A Quantitative Approach”, Morgan Kaufmann / Elsevier, 5th edition, 2012.
2. Kai Hwang, “Advanced Computer Architecture”, Tata McGraw-Hill Education, 2003.
3. Richard Y. Kain, “Advanced Computer Architecture a Systems Design Approach”, Prentice Hall, 2011.
4. David E. Culler, Jaswinder Pal Singh, “Parallel Computing Architecture: A Hardware/ Software Approach”, Morgan Kaufmann / Elsevier, 1997.

CP16111 ADVANCED DATA STRUCTURES LABORATORY

L	T	P	C
0	0	4	2

OBJECTIVES:

- To learn to implement iterative and recursive algorithms.
- To learn to design and implement algorithms using hill climbing and dynamic programming techniques.
- To learn to implement shared and concurrent objects.
- To learn to implement concurrent data structures.

LAB EXERCISES:

Each student has to work individually on assigned lab exercises. Lab sessions could be scheduled as one contiguous four-hour session per week or two two-hour sessions per week. There will be about 15 exercises in a semester. It is recommended that all implementations are carried out in Java. If C or C++ has to be used, then the threads library will be required for concurrency. Exercises should be designed to cover the following topics:

- Implementation of graph search algorithms.
- Implementation and application of network flow and linear programming problems.
- Implementation of algorithms using the hill climbing and dynamic programming design techniques.
- Implementation of recursive backtracking algorithms.
- Implementation of randomized algorithms.
- Implementation of various locking and synchronization mechanisms for concurrent linked lists, concurrent queues, and concurrent stacks.
- Developing applications involving concurrency.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

1. Design and apply iterative and recursive algorithms.
2. Design and implement algorithms using the hill climbing and dynamic programming and recursive backtracking techniques.
3. Design and implement optimisation algorithms for specific applications.
4. Design and implement randomized algorithms.
5. Design appropriate shared objects and concurrent objects for applications.
6. Implement and apply concurrent linked lists, stacks, and queues.

REFERENCES:

1. Jeff Edmonds, “How to Think about Algorithms”, Cambridge University Press, 2008.
2. M. Herlihy and N. Shavit, “The Art of Multiprocessor Programming”, Morgan Kaufmann, 2008.
3. Steven S. Skiena, “The Algorithm Design Manual”, Springer, 2008.
4. Peter Brass, “Advanced Data Structures”, Cambridge University Press, 2008.
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, “Algorithms”, McGraw Hill, 2008.
6. J. Kleinberg and E. Tardos, "Algorithm Design“, Pearson Education, 2006.
7. T. H. Cormen, C.2 E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms“, PHIL earning Private Limited, 2012.
8. Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
9. V. Aho, J. E. Hopcroft, and J. D. Ullman, “The Design and Analysis of Computer Algorithms**”,** Addison-Wesley, 1975.
10. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, ”Data Structures and Algorithms”,Pearson,2006.

CP16112	CASE STUDY: NETWORK DESIGN (Team Work)	L	T	P	C
		0	0	2	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 if config 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 if config command places a default route into the routing tables this route must be deleted. to ´blindfoldˇ 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 responsibilities 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 target host using FTP. A load is placed on the network to simulate congestion and the file is transferred, 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 n 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 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
  Describe the central and remote site configurations for three setups: central site with one inter face 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 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 as 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:

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 fire wall 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:

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.

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

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: 30 PERIODS

CP16201 THEORETICAL FOUNDATIONS OF COMPUTER SCIENCE **L T P C**
3 1 0 4

OBJECTIVES:

- To review sets, relations, functions, and other foundations
- To understand propositional and predicate logics and their applications
- To understand lambda calculus and functional programming
- To understand graph structures and their applications
- To understand formal models of computation, computability, and decidability.

UNIT I FOUNDATIONS **12**

Sets – relations – equivalence relations – partial orders – functions – recursive functions – sequences – induction principle – structural induction – recursive algorithms – counting – pigeonhole principle – permutations and combinations – recurrence relations

UNIT II LOGIC AND LOGIC PROGRAMMING **12**

Propositional logic – syntax – interpretations and models – deduction theorems – normal forms – inference rules – SAT solvers – Davis Putnam procedure – binary decision diagrams – predicate logic – syntax – proof theory – semantics of predicate logic – undecidability of predicate logic - Normal form – unification – - inferences in first-order logic – logic programming – definite programs – SLD resolution – normal programs – SLDNF resolution – introduction to Prolog.

UNIT III LAMBDA CALCULUS AND FUNCTIONAL PROGRAMMING **12**

Lambda notation for functions – syntax – curried functions – parametric polymorphism – lambda reduction – alpha reduction – beta reduction – beta abstraction – extensionality theorem – delta reduction – reduction strategies – normal forms – Church-Rosser Theorems – pure lambda calculus – constants – arithmetic – conditionals – Iteration – recursion – introduction to functional programming.

UNIT IV GRAPH STRUCTURES **12**

Tree Structures – Graph structures – graph representations – regular graph structures – random graphs – Connectivity – Cycles – Graph Coloring – Cliques, Vertex Covers, Independent sets – Spanning Trees – network flows – matching.

UNIT V STATE MACHINES **12**

Languages and Grammars – Finite State Machines – State machines and languages – Turing Machines – Computational Complexity – computability – Decidability – Church's Thesis.

TOTAL: 60 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able

- To explain sets, relations, functions
- To conduct proofs using induction, pigeonhole principle, and logic
- To apply counting, permutations, combinations, and recurrence relations
- To apply recursive functions and lambda calculus
- To explain logic programming and functional programming principles
- To apply sequential structures, tree structures, and graph structures
- To explain computational models, computability, and complexity.

REFERENCES:

1. Uwe Schoning, “Logic for Computer Scientists”, Birkhauser, 2008.
2. M. Ben-Ari, “Mathematical logic for computer science”, Second Edition, Springer, 2003.
3. John Harrison, “Handbook of Practical Logic and Automated Reasoning”, Cambridge University Press, 2009.
4. Greg Michaelson, “An introduction to functional programming through lambda calculus”, Dover Publications, 2011.
5. Kenneth Slonneger and Barry Kurtz, “Formal syntax and semantics of programming languages”, Addison Wesley, 1995.
6. Kenneth H. Rosen, “Discrete Mathematics and its applications”, Seventh Edition, Tata McGraw Hill, 2011.
7. Sriram Pemmaraju and Steven Skiena, “Computational Discrete Mathematics”, Cambridge University Press, 2003.
8. M. Huth and M. Ryan, “Logic in Computer Science – Modeling and Reasoning about systems”, Second Edition, Cambridge University Press, 2004.
9. Norman L. Biggs, “Discrete Mathematics”, Second Edition, Oxford University Press, 2002.
10. Juraj Hromkovic, “Theoretical Computer Science”, Springer, 1998.
11. J. E. Hopcroft, Rajeev Motwani, and J. D. Ullman, “Introduction to Automata Theory, Languages, and Computation”, Third Edition, Pearson, 2008.

CP16202	ADVANCED DATABASES	L	T	P	C
		3	0	0	3

OBJECTIVES:

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

UNIT I PARALLEL AND DISTRIBUTED DATABASES 9

Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Interand 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.

UNIT II OBJECT AND OBJECT RELATIONAL DATABASES 9

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.

UNIT III INTELLIGENT DATABASES 9

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 – Data log- Recursive Rules-Syntax and Semantics of Data log 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.

UNIT IV ADVANCED DATA MODELS 9

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.

UNIT V EMERGING TECHNOLOGIES 9

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

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

REFERENCES:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Fifth Edition, Pearson Education/Addison Wesley, 2007.
2. Thomas Cannolly and Carolyn Begg, "Database Systems, A Practical Approach to Design, Implementation and Management", Third Edition, Pearson Education, 2007.
3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, "Database System Concepts", Fifth Edition, McGraw Hill, 2006.
4. C.J.Date, A.Kannan and S.Swamynathan,"An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.
5. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill, Third Edition 2004.

CP16203 PRINCIPLES OF PROGRAMMING LANGUAGES

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I SYNTAX AND SEMANTICS **9**

Evolution of programming languages – describing syntax – context-free grammars – attribute grammars – describing semantics – lexical analysis – parsing – recursive-decent – bottom-upparsing.

UNIT II DATA, DATA TYPES, AND BASIC STATEMENTS **9**

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.

UNIT III SUBPROGRAMS AND IMPLEMENTATIONS **9**

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.

UNIT IV OBJECT-ORIENTATION, CONCURRENCY, AND EVENT HANDLING **9**

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.

UNIT V FUNCTIONAL AND LOGIC PROGRAMMING LANGUAGES **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

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

REFERENCES:

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

CP16204 ADVANCED OPERATING SYSTEMS L T P C

OBJECTIVES:

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

UNIT I FUNDAMENTALS OF OPERATING SYSTEMS 9

Overview – Synchronization Mechanisms – Processes and Threads - Process Scheduling – Deadlocks: Detection, Prevention and Recovery – Models of Resources – Memory Management Techniques.

UNIT II DISTRIBUTED OPERATING SYSTEMS 9

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.

UNIT III DISTRIBUTED RESOURCE MANAGEMENT 9

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 – Non blocking Commit Protocol – Security and Protection.

UNIT IV REAL TIME AND MOBILE OPERATING SYSTEMS 9

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.

UNIT V CASE STUDIES 9

Linux System: Design Principles - Kernel Modules - Process Management Scheduling – Memory Management - Input-Output Management - File System – Inter process Communication. iOS and Android: Architecture and SDK Framework - Media Layer - Services Layer - Core OS Layer – File System.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students should be able to:

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

REFERENCES:

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

CP16211 ADVANCED DATABASE LABORATORY

L	T	P	C
0	0	4	2

OBJECTIVES:

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

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, inventory Value)
   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:
   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

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 Fulltime 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.
      v. Delete person details.

   b) Extend the design by incorporating the following information.
      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].

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.
   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:

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 dataset.

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
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
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.

   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'

    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: 60 PERIODS

OUTCOMES:

- Work on distributed databases
- Create and work on object oriented databases
- Create and work with parallel database
- Experiment on active database
- Explore the features of deductive database
- To work on weka tool for clustering and classification
- Represent the database using XML and work on it

CP16212 CASE STUDY – OPERATING SYSTEMS DESIGN (Team Work) **L T P C**
0 0 2 1

OBJECTIVES:

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

LAB EXERCISES:

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.

TOTAL: 30 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

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

REFERENCES:

1. Watts S. Humphrey, “Introduction to Team Software Process”, Addison-Wesley, SEI Series in Software Engineering, 1999.
2. Mukesh Singhal and Niranjan G. Shivaratri, “Advanced Concepts in Operating Systems – Distributed, Database, and Multiprocessor Operating Systems”, Tata McGraw-Hill, 2001.
3. T. W. Doeppner, “Operating Systems in Depth: Design and Programming”, Wiley, 2010.
4. S. Tanenbaum and A. S. Woodhull, “Operating Systems Design and Implementation”, Third Edition, Prentice Hall, 2006.
5. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, Ninth Edition, John Wiley & Sons, 2012.
6. Daniel P. Bovet and Marco Cesati, “Understanding the Linux kernel”, 3rd edition, O’Reilly,2005.
7. Rajib Mall, “Real-Time Systems: Theory and Practice”, Pearson Education India, 2006.

CP16301 RECENT TRENDS IN COMPUTING TECHNOLOGIES L T P C
3 1 0 4

OBJECTIVES:

- To know the emerging trends in Android operating system.
- To have thorough knowledge of big data analytics.
- To have systematic knowledge of cloud computing.

UNIT I ANDROID OPERATING SYSTEM 9+3

Introduction-Getting started with Android programming- Activities, fragments, and intents - Getting to know the Android user interface-Designing your user interface with views-Displaying pictures and menus with views.

UNIT II ANDROID APPLICATION DEVELOPMENT 9+3

Design of android user interface with views-Using list views to display long list-understanding specialized fragments-Displaying pictures and menus-Data persistence.

UNIT III CLOUD COMPUTING 9+3

Introduction to cloud computing-Fundamentals of cloud computing -Cloud for the business-Managing the cloud resources -Advantages of highly scaled data centre - Scaling computer system -Cloud's workload strategy.

UNIT IV CLOUD ELEMENTS 9+3

Private and hybrid clouds- Infrastructure as service - Platform as service - Software as service - Massively scaled applications and business process-Managing and securing cloud services - Governing the cloud- Virtualization and the cloud.

UNIT V BIG DATA 9+3

Introduction – Drivers for Big Data – Big Data Analytics Applications – Architecture Components – Advanced Analytics Platform – Implementation of Big Data Analytics.

TOTAL (L: 45+ T:15) : 60 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in research
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability
- Identify, analyze, formulate and solve engineering problems
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.

REFERENCES:

1. Wei-Meng Lee ,Beginning Android 4 Application Development, New Delhi: Wiley Publications 2012.
2. Judith Hurwitz, Cloud computing for dummies. New Delhi: Wiley Publishing Inc.2010.
3. Michael Minelli, *Big Data Big Analytics*, New Delhi, John Wiley publications 2013.
4. Aravind Sethi, *Big data analytics Disruptive Technologies changing the Game,* IBM Corporation MC Press 2012.

CP16311	PROJECT WORK (PHASE – I)	L	T	P	C
		0	0	12	6

OBJECTIVES:

The project involves the following:

- Identification of real time problem in the field of computers
- Developing a mathematical model for solving the identified problem
- Finalization of system requirements and specification
- Proposing different solutions for the problem based on literature survey
- Future trends in providing alternate solutions
- Consolidated report preparation on the work done.

CP16411	PROJECT WORK (PHASE – II)	L	T	P	C
		0	0	24	12

OBJECTIVES:

The project involves the following:

- Preparing a project - brief proposal including
- Problem Identification
- A statement of system / process specifications proposed to be developed
- List of possible solutions including alternatives and constraints
- Cost benefit analysis
- Time Line of activities

- **A report highlighting the design finalization [based on functional requirements & standards (if any)]**

- **A presentation including the following:**
  - Implementation Phase (Hardware / Software / both)
  - Testing & Validation of the developed system
  - Learning in the Project
- **Consolidated report preparation**

CP16001 FORMAL MODELS OF SOFTWARE SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the basic elements of Z
- To understand relations, functions, and logical structures in Z
- To understand Z schemas and schema calculus
- To learn selected Z case studies
- To understand Z schema refinement

UNIT I FOUNDATIONS OF Z **9**

Understanding formal methods – motivation for formal methods – informal requirements to formal specifications – validating formal specifications – Overview of Z specification – basic elements of Z– sets and types – declarations – variables – expressions – operators – predicates and equations.

UNIT II STRUCTURES IN Z **9**

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.

UNIT III SCHEMAS AND SCHEMA CALCULUS **9**

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.

UNIT IV Z CASE STUDIES **9**

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.

UNIT V Z REFINEMENT **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Apply the basic elements of Z
- Develop relational, functional, and logical Z structures
- Develop Z schema as models of software systems
- Perform verifications and conduct proofs using Z models
- Refine Z models towards implementing software systems.

REFERENCES:

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

CP16002 PERFORMANCE EVALUATION OF COMPUTER SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I OVERVIEW OF PERFORMANCE EVALUATION **9**

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.

UNIT II MARKOV CHAINS AND SIMPLE QUEUES **9**

Discrete-Time Markov Chains – Ergodicity Theory – Real World Examples – Google, Aloha – Transition to Continuous-Time Markov Chain – M/M/1 and PASTA.

UNIT III MULTI-SERVER AND MULTI-QUEUE SYSTEMS **9**

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.

UNIT IV REAL-WORLD WORKLOADS **9**

Case Study of Real-world Workloads – Phase-Type Distributions and Matrix-Analytic Methods – Networks with Time-Sharing Servers – M/G/1 Queue and the Inspection Paradox – Task Assignment Policies for Server Farms.

UNIT V SMART SCHEDULING IN THE M/G/1 **9**

Performance Metrics – Scheduling Non-Preemptive and Preemptive Non-Size-Based Policies - Scheduling Non-Preemptive and Preemptive Size-Based Policies – Scheduling - SRPT and Fairness.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Identify the need for performance evaluation and the metrics used for it
- Discuss open and closed queueing networks
- Define Little's law and other operational laws
- Apply the operational laws to open and closed systems
- Use discrete-time and continuous-time Markov chains to model real world systems
- Develop analytical techniques for evaluating scheduling policies.

REFERENCES:

1. Mor Harchol - Balter, “Performance Modeling and Design of Computer Systems – Queueing Theory in Action”, Cambridge University Press, 2013.
2. Raj Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling”, Wiley-Interscience, 1991.
3. Lieven Eeckhout, “Computer Architecture Performance Evaluation Methods”, Morgan and Claypool Publishers, 2010.
4. Paul J. Fortier and Howard E. Michel, “Computer Systems Performance Evaluation and Prediction”, Elsevier, 2003.
5. David J. Lilja, “Measuring Computer Performance: A Practitioner’s Guide”, Cambridge University Press, 2000.
6. Krishna Kant, “Introduction to Computer System Performance Evaluation”, McGraw-Hill, 1992.
7. K. S. Trivedi, “Probability and Statistics with Reliability, Queueing and Computer Science Applications”, John Wiley and Sons, 2001.

CP16003 PROBABILISTIC REASONING SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To construct and reason with Bayesian networks
- To reason with temporal models
- To make exact and approximate inferences with graphical models
- To understand learning of parameters for probabilistic graphical models
- To understand actions and decisions with probabilistic graphical models

UNIT I REPRESENTATION **9**

Probability Theory, Graphs, Bayesian network representation: Bayes networks, Independence in graphs – Undirected graphical models: Parameterization, Markov Network independencies – Conditional Bayesian networks.

UNIT II TEMPLATE BASED REPRESENTATION **9**

Temporal models (Dynamic Bayesian networks, Hidden Markov Models) – Directed probabilistic models for object-relational domains – Inference in temporal models: Kalman filters.

UNIT III INFERENCE **9**

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.

UNIT IV LEARNING **9**

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

UNIT V ACTIONS AND DECISIONS **9**

Causality – Utilities and decisions – Structured decision problems.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Construct Bayesian networks
- Reason with Bayesian networks
- Reason with Dynamic networks and Hidden Markov Models
- Conduct inferences with Bayesian networks
- Implement algorithms to learn probabilistic graphical models
- Explain actions and decisions with probabilistic graphical models.

REFERENCES:

1. Daphne Koller and Nir Friedman, “Probabilistic Graphical Models: Principles and Techniques”, MIT Press, 2009.
2. David Barber, “Bayesian Reasoning and Machine Learning”, Cambridge University Press,2012.
3. Adnan Darwiche, “Modeling and Reasoning with Bayesian networks”, Cambridge University Press, 2009.
4. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
5. Stuart Russel and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, Prentice Hall, 2009.

CP16004 DATA ANALYSIS AND BUSINESS INTELLIGENCE L T P C
3 0 0 3

OBJECTIVES:

- To understand linear regression models
- To understand logistic regression models
- To understand generalized linear models
- To understand simulation using regression models
- To understand causal inference
- To understand multilevel regression
- To understand data collection and model understanding

UNIT I LINEAR REGRESSION 9

Introduction to data analysis – Statistical processes – statistical models – statistical inference – review of random variables and probability distributions – linear regression – one predictor – multiple predictors – prediction and validation – linear transformations – centering and standardizing – correlation – logarithmic transformations – other transformations – building regression models – fitting a series of regressions.

UNIT II LOGISTIC AND GENERALIZED LINEAR MODELS 9

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.

UNIT III SIMULATION AND CAUSAL INFERENCE 9

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 – observational studies – causal inference using advanced models – matching – instrumental variables.

UNIT IV MULTILEVEL REGRESSION 9

Multilevel structures – clustered data – multilevel linear models – partial pooling – group-level predictors – model building and statistical significance – varying intercepts and slopes – scaled inverse-Wishart distribution – non-nested models – multi-level logistic regression – multi-level generalized linear models.

UNIT V DATA COLLECTION AND MODEL UNDERSTANDING 9

Design of data collection – classical power calculations – multilevel power calculations – power calculation using fake-data simulation – understanding and summarizing fitted models – uncertainty and variability – variances – R2 and explained variance – multiple comparisons and statistical significance – analysis of variance – ANOVA and multilevel linear and general linear models – missing data imputation.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Build and apply linear regression models
- Build and apply logistic regression models
- Build and apply generalized linear models
- Perform simulation using regression models
- Perform casual inference from data
- Build and apply multilevel regression models
- Perform data collection and variance analysis.

REFERENCES:

1. Andrew Gelman and Jennifer Hill, "Data Analysis using Regression and multilevel/Hierarchical Models", Cambridge University Press, 2006.
2. Philipp K. Janert, "Data Analysis with Open Source Tools", O'Reilley, 2010.
3. Wes McKinney, "Python for Data Analysis", O'Reilley, 2012.
4. Davinderjit Sivia and John Skilling, "Data Analysis: A Bayesian Tutorial", Second Edition, Oxford University Press, 2006.
5. Robert Nisbelt, John Elder, and Gary Miner, "Handbook of statistical analysis and datamining applications", Academic Press, 2009.
6. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
7. John Maindonald and W. John Braun, "Data Analysis and Graphics Using R: An Example based Approach", Third Edition, Cambridge University Press, 2010.
8. David Ruppert, "Statistics and Data Analysis for Financial Engineering", Springer, 2011.

CP16005 IMAGE PROCESSING AND ANALYSIS

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I SPATIAL DOMAIN PROCESSING **9**

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.

UNIT II FREQUENCY DOMAIN PROCESSING **9**

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

UNIT III SEGMENTATION AND EDGE DETECTION **9**

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

UNIT IV INTEREST POINTS, MORPHOLOGY, AND TEXTURE **9**

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.

UNIT V COLOR IMAGES AND IMAGE COMPRESSION **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Explain image modalities, sensing, acquisition, sampling, and quantization
- Explain image noise models
- Implement spatial filter operations
- Explain frequency domain transformations
- Implement frequency domain filters
- Apply segmentation algorithms
- Apply edge detection techniques
- Apply corner and interest point detection algorithms
- Apply morphological operations
- Perform texture analysis
- Analyze color images
- Implement image compression algorithms

REFERENCES:

1. E. R. Davies, "Computer & Machine Vision", Fourth Edition, Academic Press, 2012.
2. W. Burger and M. Burge, "Digital Image Processing: An Algorithmic Introduction using Java", Springer, 2008.
3. John C. Russ, "The Image Processing Handbook", Sixth Edition, CRC Press, 2011.
4. R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Third Edition, Pearson,2008.
5. Mark Nixon and Alberto S. Aquado, "Feature Extraction & Image Processing for Computer Vision", Third Edition, Academic Press, 2012.
6. D. L. Baggio et al., "Mastering Open CV with Practical Computer Vision Projects", PacktPublishing, 2012.
7. Jan Erik Solem, "Programming Computer Vision with Python: Tools and algorithms for analyzing images", O'Reilly Media, 2012.

NW16002	SENSING TECHNIQUES AND SENSORS	L	T	P	C
		3	0	0	3

OBJECTIVES:

- To study the sensor characteristics and the fundamental principles of sensing
- To understand the sensor interface electronics
- To study selected motion-related sensors
- To study light and radiation detectors
- To study selected temperature sensors
- To study selected chemical sensors

UNIT I PRINCIPLES OF SENSING 9

Data Acquisition – sensor characteristics – electric charges, fields, potentials – capacitance – magnetism – inductance – resistance – piezoelectric – pyroelectric – Hall effect – thermos electric effects – sound waves – heat transfer – light – dynamic models of sensors

UNIT II OPTICAL COMPONENTS AND INTERFACE ELECTRONICS 9

Radiometry – Photometry – mirrors – lenses – fibre optics – concentrators – Interface circuits – amplifiers – light-to-voltage – excitation circuits – ADC – Digitization – Capacitance-to-voltage – bridge circuits – data transmission – noise in sensors and circuits – calibration – low power sensors

UNIT III MOTION RELATED SENSORS 9

Occupancy and motion detectors: ultrasonic – microwave – capacitive detectors – triboelectric – optoelectronic motion sensors – optical presence sensor – Pressure Gradient sensors Velocity and acceleration sensors: Accelerometer characteristics – capacitative accelerometers – piezo electric accelerometers – piezo resistive accelerometers – thermal accelerometers – Gyroscopes – piezoelectric cables – gravitational sensors

UNIT IV LIGHT AND RADIATION DETECTORS 9

Light Detectors: Photo diodes – photo transistor – photo resistor – cooled detectors – CCD and CMOS image sensors – thermal detectors – optical design – gas flame detectors Radiation Detectors: scintillating detectors – ionization detectors – cloud and bubble chambers.

UNIT V TEMPERATURE AND CHEMICAL SENSORS 9

Temperature Sensors: coupling with objects – temperature reference points – thermo resistive sensors – thermo electric contact sensors – semiconductor sensors – acoustic sensors –piezoelectric sensors Chemical sensors: characteristics – classes of chemical sensors –biochemical sensors – multi-sensor arrays – electronic noses and tongues.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Explain sensor characteristics
- Explain the physics of sensors
- Explain optical components of sensors
- Apply sensor interface electronics
- Choose and use appropriate motion-related sensors
- Choose and use appropriate light and radiation detectors
- Choose and use appropriate temperature sensors
- Choose and use appropriate chemical sensors

REFERENCES:

1. Jacob Fraden, “Handbook of Modern Sensors: Physics, Designs, and Applications”, Fourth Edition, Springer, 2010.

CP16006 **RANDOMIZED ALGORITHMS** **L T P C**
3 0 0 3

OBJECTIVES:

- To understand the mathematical foundations needed for understanding and designing randomized algorithms.
- To appreciate the need for randomized algorithms.
- To expose the students to probabilistic methods.
- To understand the concept of random walk.
- To expose the students to different types of applications of randomized algorithms.

UNIT I INTRODUCTION TO RANDOMIZED ALGORITHMS 9

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.

UNIT II PROBABILISTIC METHODS 9

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.

UNIT III ALGEBRAIC TECHNIQUES AND APPLICATIONS 9

Fingerprinting Techniques – Verifying Polynomial Identities – Perfect Matching in Graphs – Pattern Matching – Verification of Matrix Multiplication - Data Structuring Problems – Random Treaps –Skip Lists – Hash Tables.

UNIT IV GEOMETRIC AND GRAPH ALGORITHMS 9

Randomized Incremental Construction – Convex Hulls – Duality – Trapezoidal Decompositions – Linear Programming – Graph Algorithms – Min-cut – Minimum Spanning Trees.

UNIT V HASHING AND ONLINE ALGORITHMS 9

Hashing – Universal Hashing - Online Algorithms – Randomized Online Algorithms – Online Paging – Adversary Models – Relating the Adversaries – The k-server Problem.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Identify the need for randomized algorithms
- Discuss the classification of randomized algorithms
- Present the various paradigms for designing randomized algorithms
- Discuss the different probabilistic methods used for designing randomized algorithms
- Apply the techniques studied to design algorithms for different applications like matrix multiplication, hashing, linear programming.

REFERENCES:

1. Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
2. Juraj Hromkovic,”Design and Analysis of Randomized Algorithms”, Springer, 2010.
3. Michael Mitzenmacher and Eli Upfal, “Probabilty and Computing – Randomized Algorithms and Probabilistic Analysis”, Cambridge University Press, 2005.

NW16003 MOBILE AND PERVASIVE COMPUTING

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I INTRODUCTION **9**

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.

UNIT II 3G AND 4G CELLULAR NETWORKS **9**

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.

UNIT III SENSOR AND MESH NETWORKS **9**

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.

UNIT IV CONTEXT AWARE COMPUTING & WEARABLE COMPUTING **9**

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.

UNIT V APPLICATION DEVELOPMENT **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

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

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

REFERENCES:

1. Asoke K Talukder, Hasan Ahmed, Roopa R Yavagal, "Mobile Computing: Technology, Applications and Service Creation", 2nd ed, Tata McGraw Hill, 2010.
2. Reto Meier, "Professional Android 2 Application Development", Wrox Wiley,2010.
3. Pei Zheng and Lionel M Li, 'Smart Phone & Next Generation Mobile Computing', Morgan Kaufmann Publishers, 2006.
4. Frank Adelstein, 'Fundamentals of Mobile and Pervasive Computing', TMH, 2005.
5. Jochen Burthardt et al, 'Pervasive Computing: Technology and Architecture of Mobile Internet Applications', Pearson Education, 2003.
6. Feng Zhao and Leonidas Guibas, 'Wireless Sensor Networks', Morgan Kaufmann Publishers,2004.
7. Uwe Hansmaan et al, 'Principles of Mobile Computing', Springer, 2003
8. Reto Meier, "Professional Android 2 Application Development", Wrox Wiley,2010.
9. Mohammad s. Obaidat et al, "Pervasive Computing and Networking", John wiley
10. Stefan Poslad, "Ubiquitous Computing: Smart Devices, Environments and Interactions", Wiley, 2009.
11. Frank Adelstein Sandeep K. S. Gupta Golden G. Richard III Loren Schwiebert "Fundamentals of Mobile and Pervasive Computing, ", McGraw-Hill, 2005.

CP16007 PARALLEL PROGRAMMING PARADIGMS

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I FOUNDATIONS OF PARALLEL PROGRAMMING **9**

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

UNIT II MESSAGE PASSING PARADIGM **9**

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 –all gather – derived types – remote memory access – dynamic process management – MPI for grids – performance evaluation of MPI programs

UNIT III SHARED MEMORY PARADIGM: PTHREADS **9**

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

UNIT IV SHARED MEMORY PARADIGM: OPENMP **9**

Basic Open MP constructs – scope of variables – 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

UNIT V GRAPHICAL PROCESSING PARADIGMS: OPENCL AND CUDA **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Explain models of parallel programming
- Explain hardware level support for concurrency
- Explain issues in parallel programming
- Develop message-passing parallel programs using MPI framework
- Develop shared-memory parallel programs using Pthreads
- Develop shared-memory parallel programs using OpenMP
- Develop CUDA programs
- Develop OpenCL programs

REFERENCES:

1. Peter S. Pacheco, "An introduction to parallel programming", Morgan Kaufmann, 2011.
2. M. J. Quinn, "Parallel programming in C with MPI and Open MP", Tata McGraw Hill, 2003.
3. W. Gropp, E. Lusk, and R. Thakur, "Using MPI-2: Advanced features of the message passing interface", MIT Press, 1999.
4. W. Gropp, E. Lusk, and A. Skjellum, "Using MPI: Portable parallel programming with the message passing interface", Second Edition, MIT Press, 1999.
5. B. Chapman, G. Jost, and Ruud van der Pas, "Using Open MP", MIT Press, 2008.
6. D. R. Butenhof, "Programming with POSIX Threads", Addison Wesley, 1997.
7. B. Lewis and D. J. Berg, "Multithreaded programming with Pthreads", Sun Microsystems Press, 1998.
8. A. Munshi, B. Gaster, T. G. Mattson, J. Fung, and D. Ginsburg, "Open CL programming guide", Addison Wesley, 2011.
9. Rob Farber, "CUDA application design and development", Morgan Haufmann, 2011.

CP16008 SOFTWARE REQUIREMENTS ENGINEERING

L	T	P	C
3	0	0	3

OBJECTIVES:

- Understand system requirements
- Identify different types of requirement
- Generate requirements be elicitation
- Develop requirements documentation
- Evaluate the requirements.

UNIT I DOMAIN UNDERSTANDING **9**

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.

UNIT II REQUIREMENTS ELICITATION **9**

Introduction – Understanding stakeholders' needs – Elicitation techniques – interviews, questionnaire, workshop, brainstorming, prototyping – Documenting stakeholders' needs.

UNIT III FUNCTIONAL REQUIREMENTS **9**

Introduction – Features and Use cases – Use case scenarios – Documenting use cases – Levels of details – SRS documents.

UNIT IV QUALITY ATTRIBUTES AND USER EXPERIENCE **9**

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.

UNIT V MANAGING REQUIREMENTS **9**

Defining scope of the project – Context diagram – Managing requirements – Requirements properties – Traceability – Managing changes – Requirements metrics – Requirements management tools.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Define a process for requirements engineering
- Execute a process for gathering requirements through elicitation techniques.
- Validate requirements according to criteria such as feasibility, clarity, preciseness etc.
- Develop and document functional requirements for different types of systems.
- Develop and document quality attributes of the system to be implemented
- Communicate the requirements to stakeholders
- Negotiate with stakeholders in order to agree on a set of requirements.
- Detect and resolve feature interactions.

REFERENCES:

1. Axel van Lamsweerde, "Requirements Engineering", Wiley, 2009.
2. Gerald Kotonya, Ian Sommerville, "Requirements Engineering: Processes and Techniques", John Wiley and Sons, 1998
3. Dean Leffing well and Don Widrig, “Managing Software Requirements: A Use Case Approach (2nd Edition)”, Addison-wesley, 2003
4. SEI Report, “Quality Attributes Workshop”, http://www.sei.cmu.edu/library/abstracts/reports/03tr016.cfm , 200
5. J Nielsen, “Usability Engineering”, Academic Press, 1993

CP16009 SPEECH PROCESSING AND SYNTHESIS

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I FUNDAMENTALS OF SPEECH PROCESSING **9**

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.

UNIT II SPEECH SIGNAL REPRESENTATIONS AND CODING **9**

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.

UNIT III SPEECH RECOGNITION **9**

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.

UNIT IV TEXT ANALYSIS **9**

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.

UNIT V SPEECH SYNTHESIS **9**

Attributes – Formant Speech Synthesis – Concatenative Speech Synthesis – Prosodic Modification of Speech – Source-filter Models for Prosody Modification – Evaluation of TTS Systems.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

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

REFERENCES:

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

CP16010 MACHINE LEARNING TECHNIQUES

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the machine learning theory
- To implement linear and non-linear learning models
- To implement distance-based clustering techniques
- To build tree and rule based models
- To apply reinforcement learning techniques

UNIT I FOUNDATIONS OF LEARNING **9**

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

UNIT II LINEAR MODELS **9**

Linear classification – univariate linear regression – multivariate linear regression – regularized regression – Logistic regression – perceptrons – multilayer neural networks – learning neural networks structures – support vector machines – soft margin SVM – going beyond linearity – generalization and over fitting – regularization – validation.

UNIT III DISTANCE-BASED MODELS **9**

Nearest neighbor models – K-means – clustering around medoids – silhouttes – hierarchical clustering – k-d trees – locality sensitive hashing – non-parametric regression – ensemble learning– bagging and random forests – boosting – meta learning.

UNIT IV TREE AND RULE MODELS **9**

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 – association rule mining – first-order rule learning.

UNIT V REINFORCEMENT LEARNING **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- To explain theory underlying machine learning
- To construct algorithms to learn linear and non-linear models
- To implement data clustering algorithms
- To construct algorithms to learn tree and rule-based models
- To apply reinforcement learning techniques.

REFERENCES:

1. 1. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, "Learning from Data", AML Book Publishers, 2012.
2. P. Flach, "Machine Learning: The art and science of algorithms that make sense of data", Cambridge University Press, 2012.
3. K. P. Murphy, "Machine Learning: A probabilistic perspective", MIT Press, 2012.
4. C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2007.
5. D. Barber, "Bayesian Reasoning and Machine Learning", Cambridge University Press,2012.
6. M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of Machine Learning", MIT Press, 2012.
7. T. M. Mitchell, "Machine Learning", McGraw Hill, 1997.
8. S. Russel and P. Norvig, "Artificial Intelligence: A Modern Approach", Third Edition, Prentice Hall, 2009.

CP16011	CONCURRENCY MODELS	L	T	P	C
		3	0	0	3

OBJECTIVES:

- To model concurrency in FSP
- To specify and check safety and liveness properties
- To understand concurrency architectures and design
- To apply linear temporal logic to safety and liveness analysis
- To apply Petri nets for concurrency modeling and analysis

UNIT I FSP AND GRAPH MODELS **9**

Concurrency and issues in concurrency – models of concurrency – graphical models – FSP &LTSA – modeling processes with FSP – concurrency models with FSP – shared action – structure diagrams – issues with shared objects – modeling mutual exclusion – conditional synchronization– modeling semaphores – nested monitors – monitor invariants.

UNIT II SAFETY AND LIVENESS PROPERTIES **9**

Deadlocks – deadlock analysis in models – dining philosopher's problem – safety properties – single-lane bridge problem – liveness properties – liveness of the single-lane bridge – readers writers problem – message passing – asynchronous message passing models – synchronous message passing models – rendezvous.

UNIT III CONCURRENCY ARCHITECTURES AND DESIGN **9**

Modeling dynamic systems – modeling timed systems – concurrent architectures – Filter pipeline –Supervisor-worker model – announcer-listener model – model-based design – from requirements to models – from models to implementations – implementing concurrency in Java – program verification.

UNIT IV LINEAR TEMPORAL LOGIC (LTL) **9**

Syntax of LTL – semantics of LTL – practical LTL patterns – equivalences between LTL statements – specification using LTL – LTL and FSP – Fluent proposition – Temporal propositions– Fluent Linear Temporal Logic (FLTL) – FLTL assertions in FSP – Database ring problem.

UNIT V PETRI NETS **9**

Introduction to Petri nets – examples – place-transition nets – graphical and linear algebraic representations – concurrency & conflict – coverability graphs – decision procedures – liveness – colored Petri nets (CPN) – modeling & verification using CPN – non-hierarchical CPN – modeling protocols – hierarchical CPN – timed CPN – applications of Petri Nets.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Develop concurrency models and FSP
- State safety and liveness properties in FSP
- Verify properties using LTSA tool
- Explain concurrency architectures
- Design concurrent Java programs from models
- Apply Linear Temporal Logic to state safety and liveness properties
- Assert LTL properties in FSP and check using LTSA tool
- Model and analyze concurrency using Petri nets

REFERENCES:

1. 1. Jeff Magee & Jeff Kramer, "Concurrency: State Models and Java Programs", Second Edition, John Wiley, 2006.
2. M. Huth & M. Ryan, "Logic in Computer Science – Modeling and Reasoning about Systems", Second Edition, Cambridge University Press, 2004.
3. B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, and D. Lea, "Java Concurrency in Practice", Addison-Wesley Professional, 2006.
4. Wolfgang Reisig, "Petri Nets: An Introduction", Springer, 2011.
5. K. Jensen and L. M. Kristensen, "Colored Petri Nets: Modeling and Validation of Concurrent Systems", Springer, 2009.
6. Wolfgang Reisig, "Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies", Springer, 2013.

CP16012	REAL TIME SYSTEMS	L	T	P	C
		3	0	0	3

OBJECTIVES:

- To provide good understanding of fundamental concepts in real time systems.
- To provide understanding of advanced topics in real time systems.
- 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
- To expose to understand capabilities of commercial off-the-shelf R-T kernel.
- To expose to real time communications and databases.

UNIT I INTRODUCTION **9**

Real-time systems – Applications – Basic Model – Characteristics – Safety and Reliability – Real-Time tasks – Timing Constraints – Modelling Timing Constraints.

UNIT II SCHEDULING REAL-TIME TASKS **9**

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.

UNIT III RESOURCE SHARING AMONG RT TASKS & SCHEDULING RT TASKS **9**

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.

UNIT IV COMMERCIAL RT OPERATING SYSTEMS **9**

Time Services – Features of RT OS – Unix as a RT OS – Unix Based RT OS – Windows as a RTOS – POSIX – Survey of RTOS: PSOS – VRTX – VxWorks – QNX - μC/OS-II – RT Linux – Lynx –Windows CE – Benching RT Systems.

UNIT V RT COMMUNICATION & DATABASES **9**

Examples of Applications Requiring RT Communication – Basic Concepts – RT Communication ina 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.

TOTAL: 45 PERIODS

OUTCOMES:

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

REFERENCES:

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

CP16013	COMPUTER VISION	L	T	P	C
		3	0	0	3

OBJECTIVES:

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

UNIT I IMAGE PROCESSING FOUNDATIONS 9

Review of image processing techniques – classical filtering operations – thresholding techniques – edge detection techniques – corner and interest point detection – mathematical morphology – texture.

UNIT II SHAPES AND REGIONS 9

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.

UNIT III HOUGH TRANSFORM 9

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.

UNIT IV 3D VISION AND MOTION 9

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.

UNIT V APPLICATIONS 9

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Implement fundamental image processing techniques required for computer vision
- Perform shape analysis
- Implement boundary tracking techniques
- Apply chain codes and other region descriptors
- Apply Hough Transform for line, circle, and ellipse detections
- Apply 3D vision techniques
- Implement motion related techniques
- Develop applications using computer vision techniques

REFERENCES:

1. E. R. Davies, “Computer & Machine Vision”, Fourth Edition, Academic Press, 2012.
2. R. Szeliski, “Computer Vision: Algorithms and Applications”, Springer 2011.
3. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 2012.
4. Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision”, Third Edition, Academic Press, 2012.
5. D. L. Baggio et al., “Mastering Open CV with Practical Computer Vision Projects”, Packt Publishing, 2012.
6. Jan Erik Solem, “Programming Computer Vision with Python: Tools and algorithms for analyzing images”, O'Reilly Media, 2012.

NW16202	NETWORK AND INFORMATION SECURITY	L	T	P	C
		3	0	0	3

OBJECTIVES:

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

UNIT I INTRODUCTION **9**

An Overview of Computer Security-Security Services-Security Mechanisms-Security Attacks-Access Control Matrix, Policy-Security policies, Confidentiality policies, Integrity policies and Hybrid policies.

UNIT II CRYPTOSYSTEMS & AUTHENTICATION **9**

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.

UNIT III PUBLIC KEY CRYPTOSYSTEMS **9**

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.

UNIT IV SYSTEM IMPLEMENTATION **9**

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.

UNIT V NETWORK SECURITY **9**

Secret Sharing Schemes-Kerberos- Pretty Good Privacy (PGP)-Secure Socket Layer (SSL)-Intruders – HIDS- NIDS - Firewalls – Viruses.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

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

REFERENCES:

1. William Stallings, “Cryptography and Network Security: Principles and Practices”, Third Edition, Pearson Education, 2006.
2. Matt Bishop, “Computer Security art and science”, Second Edition, Pearson Education,2002.
3. Wade Trappe and Lawrence C. Washington, “Introduction to Cryptography with Coding Theory” Second Edition, Pearson Education, 2007.
4. Jonathan Katz, and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press,2007.
5. Douglas R. Stinson, “Cryptography Theory and Practice”, Third Edition, Chapman &Hall/CRC, 2006.
6. Wenbo Mao, “Modern Cryptography – Theory and Practice”, Pearson Education, First Edition, 2006.
7. Network Security and Cryptography, Menezes Bernard, Cengage Learning, New Delhi,2011.
8. Man Young Rhee, Internet Security, Wiley, 2003.
9. OWASP top ten security vulnerabilities: http://xml.coverpages.org/OWASP-TopTen.pdf.

CP16014 DESIGN AND ANALYSIS OF PARALLEL ALGORITHMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the need for parallel algorithms
- To expose the students to different models of parallel computation
- To expose the students to parallel sorting and searching algorithms
- To understand the application of the concepts studied to different types of problems
- To analyze parallel algorithms.

UNIT I INTRODUCTION **9**

Introduction to Parallel Algorithms – Models of Parallel Computation – Sorting on an EREW-SIMDPRAM Computer – Relation between PRAM Models – SIMD Algorithms – MIMD Algorithms –Selection – Desirable Properties for Parallel Algorithms - Parallel Algorithm for Selection –Analysis of Parallel Algorithms.

UNIT II SORTING AND SEARCHING **9**

Merging on the EREW and CREW Models - Fast Merging on EREW - Sorting Networks – Sorting on a Linear Array – Sorting on CRCW, CREW, EREW Models – Searching a Sorted Sequence –Searching a Random Sequence.

UNIT III ALGEBRAIC PROBLEMS **9**

Generating Permutations and Combinations in Parallel – Matrix Transpositions – Matrix by Matrix Multiplications – Matrix by Vector multiplication.

UNIT IV GRAPH THEORY AND COMPUTATIONAL GEOMETRY PROBLEMS **9**

Connectivity Matrix – Connected Components – All Pairs Shortest Paths – Minimum Spanning Trees – Point Inclusion – Intersection, Proximity and Construction Problems - Sequential Tree Traversal - Basic Design Principles – Algorithm – Analysis.

UNIT V DECISION AND OPTIMIZATION PROBLEMS **9**

Computing Prefix Sums – Applications - Job Sequencing with Deadlines – Knapsack Problem-The Bit Complexity of Parallel Computations.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Identify the need for parallel algorithms
- Discuss the classification of parallel architectures and identify suitable programming models
- Perform sorting on Sorting on CRCW, CREW, EREW Models
- Search a sorted as well as random sequence
- Develop and analyze algorithms for different applications like matrix multiplication, shortest path, job sequencing and the knapsack problem.

REFERENCES:

1. Selim G. Akl, “The Design and Analysis of Parallel Algorithms”, Prentice Hall, New Jersey,1989.
2. Michael J. Quinn, “Parallel Computing: Theory & Practice”, Tata McGraw Hill Edition, 2003.
3. Justin R. Smith, “The Design and Analysis of Parallel Algorithms”, Oxford University Press, USA , 1993.
4. Joseph JaJa, “Introduction to Parallel Algorithms”**,** Addison-Wesley, 1992.

CP16015 **SOFTWARE ARCHITECTURES** **L T P C**
3 0 0 3

OBJECTIVES:

- Understand architectural requirements
- Identify architectural structures
- Develop architectural documentation
- Generate architectural alternatives
- Evaluate the architecture against the drivers

UNIT I ARCHITECTURAL DRIVERS **9**

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.

UNIT II ARCHITECTURAL VIEWS AND DOCUMENTATION **9**

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.

UNIT III ARCHITECTURAL STYLES **9**

Introduction – Data flow styles – Call-return styles – Shared Information styles – Event styles – Case studies for each style.

UNIT IV ARCHITECTURAL DESIGN **9**

Approaches for architectural design – System decomposition – Attributes driven design – Architecting for specific quality attributes – Performance, Availability – Security – Architectural conformance.

UNIT V ARCHITECTURE EVALUATION AND SOME SPECIAL TOPICS **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Explain key architectural drivers
- Explain the influence of architecture on business and technical activities
- Identify key architectural structures
- Adopt good practices for documenting the architecture
- Develop alternative architectures for a given problem
- Explain how to use formal languages to specify architecture
- Evaluate the architecture against the drivers
- Describe the recent trends in software architecture

REFERENCES:

1. Len Bass, Paul Clements, and Rick Kazman, "Software Architectures Principles and Practices", 2n Edition, Addison-Wesley, 2003.
2. Anthony J Lattanze, "Architecting Software Intensive System. A Practitioner's Guide", Auerbach Publications, 2010.
3. Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, PauloMerson, Robert Nord, and Judith Stafford, "Documenting Software Architectures. Views and Beyond", 2nd Edition, Addison-Wesley, 2010.
4. Paul Clements, Rick Kazman, and Mark Klein, "Evaluating software architectures: Methods and case studies.", Addison-Wesley, 2001.
5. David Garlan and Mary Shaw, "Software architecture: Perspectives on an emerging discipline", Prentice Hall, 1996.
6. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, "Cloud Computing. Principles and Paradigms", John Wiley & Sons, 2011.
7. Mark Hansen, "SOA Using Java Web Services", Prentice Hall, 2007
8. David Garlan, Bradley Schmerl, and Shang-Wen Cheng, "Software Architecture-Based Self-Adaptation," 31-56.
9. Mieso K Denko, Laurence Tianruo Yang, and Yan Zang (eds.), "Autonomic Computing and Networking". Springer Verlag, 2009.

CP16016 MODEL CHECKING AND PROGRAM VERIFICATION

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand automata for model checking
- To understand LTL, CTL, and CTL*
- To understand timed automata, TCTL, and PCTL
- To understand verification of deterministic and recursive programs
- To understand verification of object-oriented programs
- To understand verification of parallel, distributed, and non-deterministic programs

UNIT I AUTOMATA AND TEMPORAL LOGICS **9**

Automata on finite words – model checking regular properties – automata on infinite words – Buchiautomata – Linear Temporal Logic (LTL) – automata based LTL model checking – Computational Tree Logic (CTL) – CTL model checking – CTL* model checking.

UNIT II TIMED AND PROBABILISTIC TREE LOGICS **9**

Timed automata – timed computational tree logic (TCTL) – TCTL model checking – probabilistic systems – probabilistic computational tree logic (PCTL) – PCTL model checking – PCTL* - Markov decision processes.

UNIT III VERIFYING DETERMINISTIC AND RECURSIVE PROGRAMS **9**

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.

UNIT IV VERIFYING OBJECT-ORIENTED AND PARALLEL PROGRAMS **9**

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.

UNIT V VERIFYING NON-DETERMINISTIC AND DISTRIBUTED PROGRAMS **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Perform model checking using LTL
- Perform model checking using CTL
- Perform model checking using CTL*
- Perform model checking using TCTL and PCTL
- Verify deterministic and recursive programs
- Verify object-oriented programs
- Verify parallel, distributed, and non-deterministic programs.

REFERENCES:

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

CP16017 EMBEDDED SOFTWARE DEVELOPMENT

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand processors and their instruction sets for embedded systems
- To understand hardware platform for embedded systems
- To design and analyze programs for embedded systems
- To design multi-tasking embedded systems with RTOS
- To understand overall embedded systems development lifecycle
- To understand distributed and multi-processor embedded systems

UNIT I PROCESSORS AND INSTRUCTION SETS **9**

Introduction to embedded computing – overview of embedded system design process – instruction sets of processors: ARM, PIC, TI C55x, TI C64x – programming I/O – modes and exceptions – coprocessors– memory system – CPU performance – CPU power consumption.

UNIT II EMBEDDED COMPUTING PLATFORM **9**

Basic computing platforms – CPU Bus – memory devices and systems – choosing a platform – development environments – debugging – consumer electronics architecture – platform-level performance analysis – design example: Audio Player.

UNIT III PROGRAM DESIGN AND ANALYSIS **9**

Components for embedded programs – models of programs – Assembly, linking, and loading – compiler optimizations – program-level performance analysis – performance optimization – program-level energy optimization – optimizing program size – program validation and testing – design example: Digital Still Camera.

UNIT IV PROCESSES AND OPERATING SYSTEMS **9**

Multiples tasks and multiple processes – multirate systems – pre-emptive RTOS – priority-based scheduling – inter-process communication – evaluating OS performance – processes and power optimization – Case study: Real-time and embedded Linux – design example: Telephone answering machine.

UNIT V SYSTEM DESIGN, NETWORKS, AND MULTIPROCESSORS **9**

System design methodologies – requirements analysis – specifications – architecture design – quality assurance – distributed embedded systems – shared-memory multiprocessors – design example: Video accelerator.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Develop assembly code for processors such as ARM, PIC Microcontroller, TI C55x, TIC64x, etc.
- Choose appropriate hardware platform for a given application
- Perform platform-level performance analysis
- Design, develop, and debug embedded programs optimized for size or performance
- Develop embedded applications using an RTOS
- Perform OS-level performance analysis
- Employ best practices in embedded software engineering
- Develop distributed embedded systems and systems with shared-memory concurrency

REFERENCES:

1. Marilyn Wolf, "Computers as Components: Principles of Embedded Computing Systems Design", Third Edition, Morgan Kaufmann, 2012.
2. Christopher Hallinan, "Embedded Linux Primer: A Practical Real-World Approach", Second Edition, Prentice Hall, 2010.
3. Karim Yaghmour et al., "Building Embedded Linux Systems", O'Reilly, 2008.
4. Arnold S. Berger, "Embedded Systems Design: An Introduction to Processes, Tools, and Techniques", CMP Books, 2001.
5. David E. Simon, "An embedded Software Primer", Addison-Wesley, 1999.

CP16018	CLOUD COMPUTING	L	T	P	C
		3	0	0	3

OBJECTIVES:

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

UNIT I CLOUD ARCHITECTURE AND MODEL **9**

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.

UNIT II VIRTUALIZATION **9**

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.

UNIT III CLOUD INFRASTRUCTURE **9**

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.

UNIT IV PROGRAMMING MODEL **9**

Parallel and Distributed Programming Paradigms – Map Reduce, Twister and Iterative Map Reduce – Hadoop Library from Apache – Mapping Applications - Programming Support -Google App Engine, Amazon AWS - Cloud Software Environments -Eucalyptus, Open Nebula, OpenStack, Aneka, Cloud Sim.

UNIT V SECURITY IN THE CLOUD **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students will be able to

- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player
- Choose the appropriate Programming Models and approach.
- Address the core issues of cloud computing such as security, privacy and interoperability
- Design Cloud Services
- Set a private cloud.

REFERENCES:

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
2. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
3. Toby Velte, Anthony Velte, Robert Elsen peter, "Cloud Computing, A Practical Approach", TMH, 2009.
4. Kumar Saurabh, "Cloud Computing – insights into New-Era Infrastructure", Wiley India,2011.
5. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud" O'Reilly
6. James E. Smith, Ravi Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
7. Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, "Grid and Cloud Computing – ABusiness Perspective on Technology and Applications", Springer.
8. Ronald L. Krutz, Russell Dean Vines, "Cloud Security – A comprehensive Guide to Secure Cloud Computing", Wiley – India, 2010.
9. Rajkumar Buyya, Christian Vecchiola, S.Tamarai Selvi, 'Mastering Cloud Computing", TMGH,2013.
10. Gautam Shroff, Enterprise Cloud Computing, Cambridge University Press, 2011
11. Michael Miller, Cloud Computing, Que Publishing,2008
12. Nick Antonopoulos, Cloud computing, Springer Publications, 2010

CP16019 DATA VISUALIZATION TECHNIQUES

L	T	P	C
3	0	0	3

OBJECTIVES:

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

UNIT I CORE SKILLS FOR VISUAL ANALYSIS 9

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.

UNIT II TIME-SERIES, RANKING, AND DEVIATION ANALYSIS 9

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.

UNIT III DISTRIBUTION, CORRELATION, AND MULTIVARIATE ANALYSIS 9

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.

UNIT IV INFORMATION DASHBOARD DESIGN I 9

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.

UNIT V INFORMATION DASHBOARD DESIGN II 9

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

- Explain principles of visual perception
- Apply core skills for visual analysis
- Apply visualization techniques for various data analysis tasks
- Design information dashboard

REFERENCES:

1. Stephen Few, "Now you see it: Simple Visualization techniques for quantitative analysis", Analytics Press, 2009.
2. Stephen Few, "Information dashboard design: The effective visual communication of data", O'Reilly, 2006.
3. Edward R. Tufte, "The visual display of quantitative information", Second Edition, Graphics Press, 2001.
4. Nathan Yau, "Data Points: Visualization that means something", Wiley, 2013.
5. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", O'Reilly, 2008.
6. Gert H. N. Laursen and Jesper Thorlund, "Business Analytics for Managers: Taking business intelligence beyond reporting", Wiley, 2010.
7. Evan Stubbs, "The value of business analytics: Identifying the path to profitability", Wiley,2011.

NW16006	PROTOCOLS AND ARCHITECTURE FOR WIRELESS SENSOR NETWORKS	L	T	P	C
		3	0	0	3

UNIT I INTRODUCTION AND OVERVIEW OF WIRELESS SENSOR NETWORKS 9

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.

UNIT II ARCHITECTURES 9

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.

UNIT III DEPLOYMENT AND CONFIGURATION 9

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.

UNIT IV ROUTING PROTOCOLS AND DATA MANIPULATION 9

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.

UNIT V SENSOR NETWORK PLATFORMS AND TOOLS 9

Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level Software Platforms, Node-level Simulators, State-centric Programming.

TOTAL: 45 PERIODS

REFERENCES:

1. 1. Holger Karl & Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks", John Wiley, 2005.
2. Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.
3. Raghavendra, Cauligi S, Sivalingam, Krishna M., Zanti Taieb, “Wireless Sensor Network”, Springer 1st Ed. 2004 (ISBN: 978-4020-7883-5).
4. Kazem Sohraby, Daniel Minoli, & Taieb Znati, “Wireless Sensor Networks- Technology, Protocols, and Applications”, John Wiley, 2007.
5. N. P. Mahalik, “Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications” Springer Verlag.
6. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

CP16020	LANGUAGE TECHNOLOGIES	L	T	P	C
		3	0	0	3

OBJECTIVES:

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

UNIT I INTRODUCTION **9**

Natural Language Processing – Mathematical Foundations – Elementary Probability Theory – Essential information Theory - Linguistics Essentials - Parts of Speech and Morphology – Phrase Structure – Semantics – Corpus Based Work.

UNIT II WORDS **9**

Collocations – Statistical Inference – n-gram Models – Word Sense Disambiguation – Lexical Acquisition.

UNIT III GRAMMAR **9**

Markov Models – Part-of-Speech Tagging – Probabilistic Context Free Grammars - Parsing.

UNIT IV INFORMATION RETRIEVAL **9**

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.

UNIT V TEXT MINING **9**

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.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the students will be able to

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

REFERENCES:

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

CP16021 AGENT BASED INTELLIGENT SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To introduce the fundamental concepts of artificial intelligence
- To explore the different paradigms in knowledge representation and reasoning
- To equip students with the knowledge and skills in logic programming using Prolog
- To recognize problems that may be solved using artificial intelligence and machine learning.

UNIT I INTRODUCTION **9**

AI Problems – Problem as a State space search – Problem characteristics - Production systems and characteristics – Issues in design of search programs – Heuristic search techniques – Hill climbing – Best-first search – Problem reduction – Constraint satisfaction.

UNIT II KNOWLEDGE REPRESENTATION **9**

Knowledge representation issues - Representation and mapping – Approaches, issues in knowledge Representation– frame problem - Using predicate logic - Simple facts in logic – Instance and ISA relationships – Computable functions and predicates – Resolution.

UNIT III UNCERTAINTY AND REASONING **9**

Symbolic reasoning under Uncertainty – Nonmonotonic reasoning – Logics – Implementation issues – Augumenting a problem-solver – Breadth first search – Depth first search implementation – Statistical reasoning – Probability and Baye's theorem – Rule-based system – Bayesian networks.

UNIT IV PLANNING AND GAME PLAYING **9**

Minimax search procedure – Adding alpha-beta cutoffs – Additional refinements – Iterative deepening – Planning – Components of planning system – Goal stack planning – Nonlinear planning using constraint posting – Hierarchical planning.

UNIT V ADVANCED TOPICS **9**

Natural language processing – Syntactic processing – Semantic analysis – Statistical NLP – Expert systems – Expert system shells – Knowledge acquisition – Prolog-The natural language of artificial intelligence –Introduction – Goals - Prolog terminology – Variables - Control structures - Arithmetic operators.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Identify, analyze, formulate and solve engineering problems.
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Acquire the knowledge of contemporary issues.

REFERENCES:

1. Elaine Rich, Kevin Knight and Shivashankar B Nair, *Artificial Intelligence*, Tata McGraw Hill, 2010.
2. Stuart Russell and Peter Norvig, *Artificial Intelligence - A Modern Approach*, Prentice Hall India, 2010.
3. M. Tim Jones, *Artificial Intelligence: A Systems Approach*, Jones and Bartlett Publisher, 2010.
4. Winston, Patrick Henry, *Artificial Intelligence*, Addison Wesley, 2008.

CP16022 KNOWLEDGE ENGINEERING L T P C
3 0 0 3

OBJECTIVES:

- To learn about knowledge representation schemes and reasoning.
- To have a thorough understanding of action planning and its representation.
- To gain skill in designing agents for the real world problems.

UNIT I INTRODUCTION 9

Key concepts –Reason for knowledge representation and reasoning – Language of first order logic – Syntax, semantics pragmatics – Expressing knowledge – Levels of representation – Knowledge acquisition and sharing – Sharing ontologies – Language ontologies –Language patterns - Tools for knowledge acquisition.

UNIT II REASONING 9

Proportional case – Handling variables and qualifies – Dealing with intractability – Reasoning with horn clauses - Procedural control of reasoning – Rules in production – Description logic - Vivid knowledge - Beyond vivid.

UNIT III REPRESENTATION AND CLASSIFICATION 9

Object oriented representations – Frame formalism – Structured descriptions – Meaning and entailment - Taxonomies and classification – Inheritance – Networks –Strategies for defeasible inheritance - Formal account of inheritance networks.

UNIT IV DEFAULTS, UNCERTAINTY AND EXPRESSIVENESS 9

Defaults – Introduction – Closed world reasoning – Circumscription – Default logic limitations of logic – Fuzzy logic – Non monotonic logic – Theories and world – Semiotics – Auto epistemic logic – Vagueness – Uncertainty and degrees of belief – Non categorical reasoning - Objective and subjective probability.

UNIT V ACTIONS AND PLANNING 9

Explanation and diagnosis – Purpose – Syntax - semantics of context – First order reasoning – Modal reasoning in context – Encapsulating objects in context – Agents – Actions – Situational calculus – Frame problem – Complex actions – Planning – Strips – Planning as reasoning - Hierarchical and conditional planning.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design and conduct experiments, as well as to analyze, interpret data on experiments relevant to Computer Science and Engineering practice.
- Identify, analyze, formulate and solve engineering problems.
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Recognize the necessity and an ability to engage in life-long learning.

REFERENCES:

1. Ronald Brachman, Hector Levesque, *Knowledge Representation and Reasoning*, Morgan Kaufmann Series in Artificial Intelligence, 2011.
2. John F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations,*Brooks Cole, 2011.
3. Arthur B. Markman, *Knowledge Representation*. United States: Lawrence Erlbaum Associates, 1998.
4. S. Simon L. Kendal, M. Creen, *An Introduction to Knowledge Engineering,* Springer, 2007.

CP16023 **BUSINESS INTELLIGENCE** **L T P C**
3 0 0 3

OBJECTIVES:

- To understand the basics of business intelligence framework.
- To learn the data integration and multi-dimensional data modeling.
- To build business intelligence applications using tools.

UNIT I INTRODUCTION OF TECHNICAL ARCHITECTURE 9

Technical architecture overview - Back room architecture - Presentation server architecture - Front room architecture – Infrastructure – Metadata - Security.

UNIT II DIMENSIONAL MODELING 9

Making the case for dimensional modeling- Dimensional modeling primer- Enterprise data warehouse bus architecture – Updates to the dimension tables - Miscellaneous dimensions - The snowflake schema - Aggregate fact tables.

UNIT III DESIGNING THE DIMENSIONAL MODELING 9

Modeling process overview - Getting organized - Four step modeling process - Design the dimensional model – Embrace data stewardship - Extract, Transform and Load overview - Extract, Transform and Load requirements and steps - Data extraction - Data transformation - Data loading.

UNIT IV BUSINESS INTELLIGENCE APPLICATIONS 9

Importance of business intelligence applications - Analytical cycle for business intelligence - Types of business intelligence applications - Navigating applications via the business intelligence portal.

UNIT V DESIGNING AND DEVELOPING BUSINESS INTELLIGENCE APPLICATIONS 9

Business intelligence application resource planning - Business intelligence application specification – Business intelligence application development - Business intelligence application maintenance.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability.
- Understand the impact of engineering solutions in a global, economic, environmental and societal context.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.

REFERENCES:

1. Raiph Kimball-Ross, *The Data Warehouse Lifecycle Toolkit*, Wiley Publication, 2008.
2. Ponniah, *Data Warehousing Fundamental*, Wiley Publication, 2010.
3. Anahory and Murray, *Data Warehousing in the Real World,* Pearson Education India, 2004.
4. Inmon, W. H. *Building the Data Warehouse*, Wiley Publication, 2005.

CP16024 NATURAL LANGUAGE PROCESSING

L	T	P	C
3	0	0	3

OBJECTIVES:

- To understand the use of state automata for language processing
- To provide the fundamentals of parsing
- To pioneer the basics of Information Retrieval (IR).

UNIT I INTRODUCTION **9**

Knowledge in speech and language processing – Ambiguity – Models and algorithms – Language, Thought and understanding – History of NLP.

UNIT II WORDS **9**

Regular expressions and automata – Words and transducers – N- grams – Part-of-speech tagging – Hidden markov and maximum entropy models.

UNIT III SPEECH **9**

Phonetics – Speech synthesis – Automatic speech recognition – Advanced topics on speech recognition – Computational phonology.

UNIT IV SYNTAX, SEMANTICS AND PRAGMATICS **9**

Formal grammars of English – Syntactic parsing – Statistical parsing – Features and unification – Language and complexity – Representation of meaning – Computational semantics – Lexical semantics.

UNIT V APPLICATIONS **9**

Information extraction – Question answering - Summarization – Dialog agents – Conversational agents – Machine translation.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Recognize the necessity and an ability to engage in life-long learning
- Acquire the knowledge of contemporary issues.

REFERENCES:

1. Daniel Jurafsky and James H. martin, *Speech and Language Processing* , Pearson Education, 2008.
2. David A. Grossman and Ophir Fedier, *Information Retrieval: Algorithms and Heuristics (The Information Retrieval Series)*, Springer, 2004.
3. Michael W Berry, *Survey of Text mining I: Clustering, Classification and Retrieval*, Copyrighted material, 2004.
4. Daniel Bikel and Imed Zitouni, *Multilingual Natural Language Processing Applications: From Theory to Practice*, IBM Press, 2012

CP16025 **SECURITY IN COMPUTING** **L T P C**
3 0 0 3

OBJECTIVES:

- To learn the vulnerabilities and threats in the computing systems.
- To understand the cryptography based approaches in security.
- To know the implementation and use of security mechanisms.

UNIT I SECURITY ISSUES IN COMPUTING 9

Introduction to computer security – Computer criminals – Methods of defense – Cryptography- Substitution ciphers – Transpositions – Making encryption algorithms – Data encryption standard – Public key encryption- Possible attacks on RSA - Uses of encryption.

UNIT II PROGRAM SECURITY AND NETWORK SECURITY 9

Secure programs – Non-malicious program errors – Targeted malicious code- Trojans – Trapdoors – Salami attack – Keystroke logging– Man-in-the middle attack- Covert channels- Controls against program threats- Threats in networks – Network security controls - Firewalls – Intrusion detection system.

UNIT III DATABASE SECURITY AND DATA MINING SECURITY 9

Introduction to databases - Security requirements – Reliability and integrity – Sensitive data – Inference – Multilevel database-Proposals for multilevel security – Data mining- Privacy and sensitivity – Data correctness and integrity – Availability of data- Privacy concepts-Privacy principles and policies - Authentication and privacy- Privacy preserving data mining.

UNIT IV DESIGN AND PROTECTION OF OPERATING SYSTEM 9

Protected objects and methods of protection- Memory and address protection- Control of access to general objects- File protection mechanisms – User authentication- Trusted system – Security policies – Models of security- Trusted operating system design- Assurance methods in trusted operating systems.

UNIT V LEGAL ISSUES IN COMPUTER SECURITY 9

Protecting programs and data – Information and the law – Rights of employees and employers – Computer crime- Administering security- Security planning – Risk analysis – Organizational security policies- Physical security.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Identify, analyze, formulate and solve engineering problems.
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Use the techniques, skills and modern engineering tools necessary for engineering practice.

REFERENCES:

1. Charles B.fleeger and Shari Lawrence Pfleeger, *Security in Computing*, Pearson Education, 2009.
2. William Stallings, *Cryptography and Network Security: Principles and Practice,* Prentice Hall of India/Pearson Education, New Delhi, 2007.
3. Dieter Gollmann, *Computer Security,* John Wiley & Sons Ltd., 2011.
4. Douglas R.Stinson, *Cryptography Theory and Practice*, CRC, 2006.

CP16026 ADHOC AND SENSOR NETWORKS L T P C
3 0 0 3

OBJECTIVES:

- To gain knowledge on routing and protocols in adhoc and sensor networks.
- To get skilled in wireless networks technology platforms and standards.
- To learn real time traffic support in wireless networks with working principles of wireless LAN.
- To get familiar in standards of wireless LAN and learn hybrid networks.

UNIT I ADHOC WIRELESS NETWORKS AND MAC 9

Introduction – Issues in ad Hoc wireless networks- MAC protocols – Issues, classifications of MAC protocols, Contention based protocols - Contention based protocols with reservation mechanism- Multi channel CSMA and power control MAC protocol.

UNIT II ROUTING PROTOCOLS AND TCP OVER AD HOC 9

Issues in designing a routing protocol – Classifications of routing protocols – Hierarchical and power aware. multicast routing –Classifications- Tree based- Mesh based adhoc transport layer issues- TCP over adhoc – Feedback based - TCP with explicit link- TCP-Bus - adhoc TCP and split TCP- adhoc transport protocol.

UNIT III QUALITY OF SERVICE IN AD HOC WIRELESS NETWORKS 9

Real-time traffic support – Issues and challenges in providing QoS – Classification of QoS solutions– MAC layer solutions – QoS routing protocols – Ticket based and predictive location based QoS routing protocols- On-Demand link state multipath QoS routing protocol- QoS frameworks- Energy management adhoc – Battery and power management schemes - Transmission power management schemes.

UNIT IV WIRELESS SENSOR NETWORKS 9

Introduction – Sensor network architecture- Data dissemination – Gathering- MAC protocols for sensor networks – Self organizing- Hybrid TDMA/FDMA and CSMA based MAC - Location discovery and quality of sensor networks-Evolving standards - Energy efficient design.

UNIT V HYBRID WIRELESS NETWORKS 9

Introduction- Next generation hybrid wireless architectures-Routing in hybrid wireless networks- Power control schemes and load balancing in hybrid wireless networks- Recent advances in wireless networks –Ultra wide band radio communication-Wireless fidelity systems-Optical wireless networks.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design a system, component or process to meet desired needs within realistic constraints such as economic,
- environmental, social, ethical, manufacturability and sustainability.
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Recognize the necessity and ability to engage in life-long learning.

REFERENCES:

1. C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc Wireless Networks – Architectures and Protocols*, New Delhi: Pearson Education, 2004.
2. Feng Zhao and Leonidas Guibas, *Wireless Sensor Networks*. Noida: Morgan Kaufman Publishers, 2004.
3. C. K. Toh, *Ad Hoc Mobile Wireless Networks.* New Delhi: Pearson Education, 2002.
4. Thomas Krag and Sebastin Buettrich, *Wireless Mesh Networking*. Mumbai: O'Reilly Publishers,2007

CP16027 HIGH PERFORMANCE COMPUTING L T P C
3 0 0 3

OBJECTIVES:

- To know how modern high performance processors are organized their strengths and weaknesses.
- To study about the architecture of parallel systems.
- To gain depth knowledge about the analytical parallel algorithms.

UNIT I MODERN PROCESSORS 9

Stored-program computer architecture – General-purpose cache-based microprocessor architecture – Memory hierarchies - Multicore processors - Multithread processors - Vector processors - Basic optimization techniques for serial code - Common sense optimizations - Simple measures - large impact - Role of compilers.

UNIT II PARALLEL COMPUTERS 9

Data access optimization - Balance analysis and lightspeed estimates - Storage order - Taxonomy of parallel computing paradigms - Shared memory computers - Distributed memory computers - Hierarchical systems – Networks - Basics of parallelization- Parallelism – Parallel scalability.

UNIT III INTRODUCTION TO PARALLEL COMPUTING 9

Motivating parallelism - Scope of parallel computing - Parallel programming platforms: Implict parallelism trends in microprocessor architectures - Limitations - Dichotomy - Physical organizations - Communication costs – Routing mechanisms for interconnected networks- Impact of process.

UNIT IV PRINCIPLES OF PARALLEL ALGORITHM DESIGN 9

Preliminaries - Decomposition techniques - Characteristics of tasks and interactions - Mapping techniques for load balancing - Methods for containing interaction overheads - Parallel algorithm models – Basic communication operations.

UNIT V SORTING AND GRAPH ALGORITHMS 9

Dense matrix Algorithm: Matrix-vector multiplication - Martix- matrix multiplication- Issues in sorting on parallel computing - Sorting networks - Bubble sorts and its variants - Quick sort - Graph algorithms - Definition and representation - Prims algorithm - Dijkstra's algorithm - All pairs shortest path - Transitive closure – Connected components.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability.
- Identify, analyze, formulate and solve engineering problems.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.

REFERENCES:

1. Georg Hager and Gerhard Wellein, *Introduction to High Performance Computing for Scientists and Engineers,* Chapman & Hall, 2010.
2. Ananth Grama and George Karypis, *Introduction to parallel computing*, Addison-Wesley 2009.
3. John Levesque and Gene Wagenbreth, *High Performance Computing: Programming and Applications*, Chapman & Hall, 2010
4. John L. Hennessy and David Patterson, *Computer Architecture- A Quantitative Approach*, Elsevier, 2012.

CP16028 EMBEDDED REAL TIME OPERATING SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To know the hardware and software of embedded systems.
- To learn the embedded programming concepts in Linux.
- To understand the real time operating systems and inter-task communication.

UNIT I INTRODUCTION AND ARCHITECTURE OF EMBEDDED SYSTEMS 9

Embedded systems – Application areas – Categories – Overview of embedded system architecture – Specialties - Recent trends in embedded systems – Hardware architecture – Software architecture – Application software – Communication software – Process of generating executable image.

UNIT II EMBEDDED SYSTEM DEVELOPMENT PROCESS 9

Development process – Requirements engineering – Design – Implementation – Integration and testing – Packaging – Configuration management – Managing embedded system development projects – Hardware platforms – Communication interfaces-Need for communication interfaces – RS232/UART – USB – Infrared – IEEE 1394 Firewire – Ethernet – IEEE 802.11.

UNIT III EMBEDDED/REAL TIME OPERATING SYSTEM CONCEPTS 9

Architecture of kernel – Tasks and task scheduler – Interrupt service routines – Semaphores - Mutex – Mailboxes – Message queues – Event registers – Pipes – Signals – Timers – Memory management – Priority inversion problem – Off the shelf operating systems – Embedded operating systems – Real time operating systems – Programming for embedded systems – Programming in linux.

UNIT IV EMBEDDED SYSTEM APPLICATIONS 9

Representative embedded systems - RFID systems – DSP embedded systems- Need for DSP based embedded systems – Overview of digital signal processing – Applications of DSP – Digital signal processor architecture – DSP based embedded system design process.

UNIT V EMBEDDED SOFTWARE DEVELOPMENT AND FUTURE TRENDS 9

Embedded software development: 89C51 Micro-controller platform – AVR Micro-controller platform – Intel String ARM platform – Future trends- Pervasive/ ubiquitous computing – Java for embedded systems - Security of embedded systems.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design and conduct experiments as well as to analyze, interpret data on experiments relevant to Computer Science and Engineering practice
- Understand the impact of engineering solutions in a global, economic, environmental and societal context.
- Recognize the necessity and ability to engage in life-long learning.

REFERENCES:

1. K.V.K.K. Prasad, Embedded/Real-Time Systems: Concepts, Design and Programming, Black Book, Dreamtech Press, 2010.
2. Rajib Mall, Real-time Systems: Theory and Practice, Pearson Education India, 2007.
3. Phillip A. Laplante, Real- Time Systems Design and Analysis, John Wiley and Sons, 2006.
4. Iyer and Gupta, Embedded Real Time Systems Programming, Tata McGraw-Hill Education, 2003.

CP16029 HARDWARE AND TROUBLE SHOOTING

L	T	P	C
3	0	0	3

OBJECTIVES:

- To learn the characteristics of hardware devices
- To understand the troubleshooting fundamentals of computer hardware and software
- To know the installation, assembling and configuration management of computer hardware and software.

UNIT I INTRODUCTION **9**

Introduction - Computer organization – Number systems and codes – Memory – Arithmetic and Logic Unit – Control unit –Instruction prefetch – Interrupts – Input/Output techniques – Device controllers - Error detection techniques – Microprocessor – Personal computer concepts – Advanced system concepts –Microcomputer concepts – Operating system – Multitasking and multiprogramming – Virtual memory – Cache memory – Modern PC and user.

UNIT II PERIPHERAL DEVICES **9**

Introduction – Keyboard – CRT display monitor – Printer – Magnetic storage devices – Floppy disk drive – Hard disk drive – Special types of disk drives – Mouse and trackball – Modem – Fax modem – CD ROM drive – Scanner – Digital camera – Digital versatile disk – Special peripherals.

UNIT III PC HARDWARE OVERVIEW **9**

Introduction – Hardware BIOS DOS interaction – The PC family – PC hardware – Inside the system box – Motherboard logic – Memory space – Peripheral interfaces and controllers – Keyboard interface – CRT display controller– Floppy disk controller – Hard disk controller.

UNIT IV INSTALLATION AND PREVENTIVE MAINTENANCE **9**

Introduction – System configuration – Pre installation planning – Installation practice – Routine checks – PC assembling and integration – BIOS setup – Engineering versions and compatibility – Preventive maintenance – Disk operating system – Virus – Data recovery.

UNIT V TROUBLESHOOTING **9**

Introduction – Computer faults – Nature of faults – Types of faults – Diagnostic programs and tools – Microprocessor and firmware – Programmable LSI's – Bus faults – Faults elimination process – Systematic troubleshooting – Symptoms observation and analysis – Fault diagnosis – Fault rectification – Troubleshooting levels – Serial port problems – FDC problems –HDC problems –Display adapter problems -FDD,HDD,CD-ROM problems.

TOTAL: 45 PERIODS

OUTCOMES:

- Design and conduct experiments as well as to analyze, interpret data on experiments relevant to Computer Science and Engineering practice.
- Use the techniques, skills and modern engineering tools necessary for engineering practice.

REFERENCES:

1. B. Govindarajalu, *IBM PC Clones Hardware, Troubleshooting and Maintenance*, TMH, 2008.
2. Jean Andrews, *A+ Guide to Managing and Maintaining Your PC*, Cengage Learning, 2013.
3. Peter Abel, Niyaz Nizamuddin, *IMB PC Assembly Language and Programming*, Pearson Education, 2007.
4. Michael Meyers, *Introduction to PC Hardware and Troubleshooting*, McGraw Hill, 2003.

CP16030 **SOFTWARE TESTING** **L T P C**
3 0 0 3

OBJECTIVES:

- To learn the different types of testing strategies.
- To know graph matrices and its applications.
- To understand the various methods to improve software testing.

UNIT I INTRODUCTION **9**

Purpose of testing- Dichotomies- Model for testing- The taxonomy of bugs: Consequences of bugs- Taxonomy of bugs.

UNIT II FLOW GRAPHS AND PATH TESTING **9**

Path-testing basics- Predicates, path predicates and achievable paths- Path sensitizing- Path instrumentation- Implementation and application of path testing.

UNIT III TRANSACTION AND DATA FLOW TESTING **9**

Transaction flows- Transaction flow testing techniques- Implementation Comments- Dataflow testing basics – Data flow testing strategies – Application, tools and effectiveness.

UNIT IV DOMAIN AND LOGIC BASED TESTING **9**

Domains and paths- Nice and ugly domains- Domain testing- Domains and interface testing- Domains and testability - Logic based testing: Decision tables- Path expressions again- KV charts.

UNIT V GRAPH MATRICES AND APPLICATION **9**

Motivational overview- Matrix of graph- Relations- Power of a matrix- Node reduction algorithm- Building tools.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Define, assess, tailor the software quality practices and software processes with methodologies for appropriate application on software development projects in various domain areas.
- Use the techniques, skills and modern Engineering tools necessary for engineering practice.

REFERENCES:

1. Boris Beizer, Software Testing Techniques,Dreamtech,2009.
2. William Perry, Software Testing : Effective Methods for Software Testing, John Wiley, 2009.
3. Cem Kaner, Jack Falk, Hung Quoc Nguyen, Testing Computer Software, Thomson Computer Press.
4. Roger S. Pressman , Software Engineering – A practitioner’s approach, Tata McGraw Hill,2011.

CP16031 **DESIGN PATTERNS** **L T P C**
3 0 0 3

OBJECTIVES:

- To understand the knowledge about patterns.
- To design patterns that enable the reuse of software architectures.
- To investigate the development of good design patterns.

UNIT I BACKGROUND ON DESIGN PATTERNS 9

Pattern concept - Pattern taxonomy - Design structures - Design principles- The singleton classifying patterns - Design patterns - The learning process - Studying design patterns - Object oriented approaches - The java foundation classes - Java design patterns - The development challenge.

UNIT II DESIGN PATTERN CATALOG 9

Creational patterns - Factory pattern - Abstract factory pattern - Builder pattern - Factory method – Prototypeton pattern - Summary of creational patterns.

UNIT III THE JAVA FOUNDATION CLASSES 9

Installing and using the JFC - Ideas behind swing - The swing class hierarchy - Writing a simple JFC program - Buttons and toolbars -Menus and actions - The JList class - The JTable class - The JTree class.

UNIT IV STRUCTURAL PATTERNS 9

Adapter - Bridge – Composite – Decorator – Façade – Flyweight – Proxy - Comparison with structural patterns.

UNIT V BEHAVIORAL PATTERNS 9

Chain of responsibility- Command - Interpreter – Iterator – Mediator – Memento – Observer – State – Strategy - Template method – Visitor - Discussion of behavioral patterns.

TOTAL: 45 PERIODS

OUTCOMES:

- Design and conduct experiments, as well as to analyze, interpret data on experiments relevant to Computer Science and Engineering practice.
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability.
- Define, assess, and tailor software quality practices, and software processes and methodologies for appropriate application on software development projects in a variety of domain areas
- Recognize of the need for, and an ability to engage in life-long learning.

REFERENCES:

1. James W. Cooper, *The Design Patterns Java Companions*, Addision Wesley Design Patterns Series, 2012.
2. Bruce Eckel, *Thinking in Patterns with Java*, MindView Inc, 2006.
3. Dr. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson publications Ltd, 2004.
4. Cay S. Horstmann , *Object-Oriented Design and Patterns*, John Wiley & Sons, 2005.

CP16032 HUMAN COMPUTER INTERACTION

L	T	P	C
3	0	0	3

OBJECTIVES:

- To Learn the design techniques and fundamentals of Human Computer Interaction (HCI)
- To know the various types of existing interfaces and evaluation techniques
- To understand the applications of HCI in emerging trends.

UNIT I INTERACTION DESIGN AND CONCEPTUALIZING INTERACTION 9

Introduction – Good and poor design – User experience – The process of interaction design – Interaction design and the user experience – Understanding the problem space and conceptualizing design – Conceptual models – Interface metaphors – Interaction types – Paradigms, theories, models and frameworks.

UNIT II SOCIAL, EMOTIONAL INTERACTION AND INTERFACES 9

Introduction – Face-to-Face conversations – Remote conversations – Telepresence – Co-presence – Emergent social phenomena - Emotional interaction: Introduction – Emotions and the user experience – Expressive interfaces – Frustrating interfaces - Persuasive technologies and behavioral change – Anthropomorphism and zoomorphism – Models of emotion – Interface types.

UNIT III DATA GATHERING, ANALYSIS, INTERPRETATION AND PRESENTATION 9

Introduction – Five key issues – Data recording – Interviews – Questionnaires –Observation – Choosing and combining Techniques- Qualitative and quantitative – Simple quantitative analysis – Simple qualitative analysis – Tools to support data analysis.

UNIT IV EVALUATION FRAMEWORK 9

Goals of evaluation – Types of evaluation – DECIDE: A framework to guide evaluation – Usability testing – Conducting experiments – Field studies – Inspections- Heuristic evaluation and walkthroughs – Analytics – Predictive models.

UNIT V UBIQUITOUS COMPUTING, HYPERTEXT AND WORLD WIDE WEB 9

Ubiquitous computing application research – virtual & augmented reality –Understanding hypertext – finding things – Web technology and issues – Static web content – Dynamic web content- Groupware systems – Computer mediated communication – DSS – Frameworks for groupware- Information and data visualization.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Identify, analyze, formulate and solve engineering problems
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.

REFERENCES:

1. Yvonne Rogers, Helen Sharp, Jenny Preece, *Interaction Design: beyond human-computer interaction*, John-Wiley and Sons Inc., 2011.
2. Alan Dix , Janet Finlay, Gregory D.Abowd, Russell Beale, *Human Computer Interaction*, Pearson Education, 2008.
3. Jonathan Lazar Jinjuan, Heidi Feng, Harry Hochheiser, *Research Methods in Human-Computer Interaction,* Wiley, 2010.
4. Dov Te'eni, Jane Carey, Ping Zhang, *Human-Computer Interaction: Developing Effective Organizational Information Systems*, John-Wiley and Sons Inc., 2007.

CP16033 MOBILE OPERATING SYSTEMS

L	T	P	C
3	0	0	3

OBJECTIVES:

- To know the emerging trends in mobile operating systems.
- To have systematic knowledge of Symbion and Mac OS X operating system.
- To have thorough knowledge of Android operating systems concepts.

UNIT I ANDROID OPERATING SYSTEM 9

Android overview-The stack-Installing android SDK-Main building blocks-Android user interface-Preferences, file systems, option menu and intents.

UNIT II ANDROID APPLICATION DEVELOPMENT 9

Design of android user interface with views-Using list views to display long list-Understanding specialized fragments-Displaying pictures and menus-Data persistence.

UNIT III WINDOWS MOBILE OPERATING SYSTEM 9

Introduction to Windows Mobile 8– Working with Apps- Customizing Windows 8- Surfing the web-Configuring the tablet-Implementing security –Maintaining Windows 8.

UNIT IV MAC AND IOS 9

The Evolution of OSX- OS X Versions-iOS-The future OS X-Architectural overview-User experience layer –Unix core –System directories-Applications-Frameworks-System calls- High level view of XNU.

UNIT V SYMBION OPERATING SYSTEM 9

The background of Symbion OS- History- Architecture of Symbion OS-OS Layered model-UI Framework layer- The application services layer.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability.
- Identify, analyze, formulate, and solve engineering problems
- Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
- Use the techniques, skills, and modern engineering tools necessary for engineering practice.

REFERENCES:

1. Marko Gargenta, *Learning Android, New* Delhi: O'Reilley 2011.
2. Wei-Meng Lee, *Beginning Android 4 application development,* New Delhi: John Wiley 2012
3. Paul McFedrics. *Teach yourself Microsoft Windows 8 Tablets,* New Delhi: John Wiley 2012
4. Jonathan Levin,*Mac OS X and iOS Internal,.* New Delhi: John Wiley 2012.

CP16034 RESEARCH METHODOLOGY

L	T	P	C
3	0	0	3

OBJECTIVES:

- To gain insights into how scientific research is conducted
- To learn and understand the optimization methods
- To identify the influencing factors or determinants of research parameters
- To prepare the documentation of research results.

UNIT I INTRODUCTION **9**

The nature of CS research - what is research? - Project planning, tools and techniques for planning – Literature searches, information gathering.

UNIT II PROJECT DEVELOPMENT **9**

Reading and understanding research papers - Project implementation and IT project management. – Presentation skills, written and oral - Time management- Team working.

UNIT III OPTIMIZATION METHODS **9**

Linear Programming: Simplex method – Dynamic Programming – Integer Programming - Hill climbing.

UNIT IV ADVANCED OPTIMIZATION TECHNIQUES **9**

Simulated annealing - Quantum annealing - Genetic algorithms - Ant colony optimization - Particle swarm optimization - Tabu search - Beam search.

UNIT V ISSUES AND TECHNICAL WRITING

Commercial and economic considerations in the IT industry - Review of Legal, Ethical, Social and Professional (LSEP) issues, such as data protection, hacking, etc. - Technical writing, referencing, bibliographies.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in research.
- Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, ethical, manufacturability, and sustainability.
- Identify, analyze, formulate and solve engineering problems.

REFERENCES:

1. C. W. Dawson, *The Essence of Computer Projects: A Student Guide.* New Delhi: PHI, 2006.
2. Duane A. Bailey, *A Letter to Research Students*. Massachusetts.
3. Humdy Taha, *Operation Research*. New Delhi: PHI, 2007.
4. S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi. *Optimization by Simulated Annealing, Science*, Vol 220, 1983, 671-680.
5. B. Apolloni, N. Caravalho and D. De Falco. *Quantum stochastic optimization, Stochastic Processes and their Applications,* Vol. 33, 1989, 233-244.
6. David E. Goldberg. *Genetic Algorithms in Search, Optimization*, *and Machine Learning*, New Delhi : New Age, 1989.

CP16036	SOFT COMPUTING	L	T	P	C
		3	0	0	3

OBJECTIVES:

- To become familiar with various Soft Computing Techniques
- To introduce different evolutionary and swarm algorithms
- To bring in the ideas of fuzzy sets, fuzzy logic and use of heuristics.

UNIT I INTRODUCTION TO SOFT COMPUTING AND NEURAL NETWORKS 9

Introduction - Soft computing constituents – From conventional AI to computational intelligence – Evolutionary computation – Neuro-Fuzzy and soft computing characteristics.

UNIT II GENETIC ALGORITHMS 9

Introduction to Genetic Algorithm (GA) – Goals of optimization – Simple GA – Simulation – Important similarities - Applications of GA – Rise of GA - GA application of historical interest – Improvements in basic technique – De Jong and function optimization.

UNIT III NEURAL NETWORKS 9

Adaptive networks – Back propagation for feed forward networks – Batch learning – Pattern by pattern learning - Supervised learning neural networks – Radial basis function networks – Unsupervised learning neural networks – Competitive learning network – Kohonen self organising networks- Hebbian learning.

UNIT IV FUZZY LOGIC 9

Fuzzy sets – Set theoretic operations – Fuzzy rules and fuzzy reasoning – Extension principle and fuzzy relation – Fuzzy If-then rules - Fuzzy inference systems – Mamdani fuzzy models – Sugeno fuzzy models – Tsukamoto fuzzy models.

UNIT V NEURO-FUZZY MODELING 9

Adaptive neuro-fuzzy Inference systems – Classification and regression trees – Decision trees – CART algorithm for tree induction - Data clustering algorithms.

TOTAL: 45 PERIODS

OUTCOMES:

- Apply basic principles and practices of Computer Science and Engineering to productively engage in the research.
- Recognize the necessity and ability to engage in life-long learning
- Acquire the knowledge of contemporary issues.

REFERENCES:

1. Jyh-Shing Roger Jang, Chuen-Tsai Sun and Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A
2. Computational Approach to Learning and Machine Intelligence, New Delhi: Prentice-Hall of India, 2003.
3. David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Singapore: Addison Wesley, 2001.
4. James A. Freeman and David M. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques. New Delhi: Pearson Education, 2003.
5. Mitchell Melanie, An Introduction to Genetic Algorithm. New Delhi: Prentice Hall, 1998.
6. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic-Theory and Applications. New Delhi: PHI 1995.
7. Jacek M. Zurada, Introduction to Artificial Neural Systems. Boston: PWS Publishers, 1992