

COURSE OBJECTIVES

- To understand the fundamental data structures and algorithms.
- To analyze theoretical analysis, and application of data structure.
- To learn linear data structures such as list, stack, and queue.
- To understand the methods of sorting, searching, hashing of algorithms
- To analyze the algorithms and their complexity

UNIT I LINEAR STRUCTURES**9**

Abstract Data Types (ADT) - List ADT -array-based implementation -linked list implementation - cursor- based linked lists -doubly linked lists -applications of lists - Stack ADT - Queue ADT -circular queueimplementation -Applications of stacks and queues

UNIT II TREE STRUCTURES**9**

Tree ADT -tree traversals -left child right sibling data structures for general trees -Binary Tree ADT - expression trees - applications of trees -binary search tree ADT- AVL trees – binary heaps.

UNIT III HASHING AND SETS**9**

Hashing –Separate Chaining – Open Addressing – Rehashing – Extendible Hashing – - Disjoint Set ADT - dynamic equivalence problem -smart union algorithms - path compression –applications of Sets.

UNIT IV GRAPHS**9**

Definitions - Topological sort - breadth-first traversal - shortest-path algorithms - minimum spanning tree - Prim's and Kruskal's algorithms -Depth-first traversal – bi connectivity -Euler circuits -applications of graphs

UNIT V ALGORITHM DESIGN AND ANALYSIS**9**

Introduction to algorithm design techniques: Greedy algorithms, Divide and conquer, Dynamic programming, backtracking, branch and bound, Randomized algorithms – Introduction to algorithm analysis: asymptotic notations, recurrences – Introduction to NP-complete problems

TOTAL: 45 PERIODS**COURSE OUTCOMES**

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

- gain knowledge in the various data structure concepts.
- implement abstract data types for linear data structures.
- apply the different linear data structures and find appropriate solutions for the problem.
- review various implementations and operation of priority queue.
- implement the concept of depth first search and bi-connectivity

TEXT BOOK

1. M. A. Weiss, "Data Structures and Algorithm Analysis in C", Fourth Edition, Pearson Education, 2013.

REFERENCES

1. V. Aho, J. E. Hopcroft, and J. D. Ullman, 'Data Structures and Algorithms', Pearson Education, 2009.
2. R. F. Gilberg, B. A. Forouzan, 'Data Structures', Second Edition, Thomson India Edition, 2008.
3. M. Tenenbaum, Y. Langsam, and M. J. Augenstein, 'Data Structures using C', Pearson Education, 2007.
4. K.S. Easwarakumar, Object Oriented Data Structures using C++, Vikas Publishing House pvt.Ltd., 2010.
5. Sara Baase and A. Van Gelder, 'Computer Algorithms', Third Edition, Pearson Ed., 2011

WEB LINKS

1. <http://www.nptel.ac.in/>
2. <http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms>
3. <https://www.youtube.com/watch?v=RpRRUQFbePU>

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	3	3	-	-	-	-	-	-	-	3	3	3
CO2	3	2	3	3	-	-	-	-	-	-	-	2	3	3
CO3	2	3	1	-	-	-	1	-	-	-	-	2	3	3
CO4	3	3	3	3	-	-	-	-	-	-	-	2	1	3
CO5	3	3	3	-	-	2	-	-	-	-	-	3	3	2



COURSE OBJECTIVES

- To be familiar with c programming
- To implement abstract data types
- To implement singly and doubly linked lists.
- To implement sorting and searching algorithms.

LIST OF EXPERIMENTS

1. Implement singly and doubly linked lists.
2. Represent a polynomial as a linked list and write functions for polynomial addition.
3. Implement stack and use it to convert infix to postfix expression
4. Implement array-based circular queue and use it to simulate a producer-consumer problem.
5. Implement an expression tree. Produce its pre-order, in-order, and post-order traversals.
6. Implement binary search tree.
7. Implement priority queue using heaps
8. Implement hashing techniques.
9. Implement Dijkstra's algorithm using priority queues
10. Implement Greedy algorithm using C.
11. Implement Branch and bound, Divide and Conquer algorithm using C.

TOTAL PERIODS: 30**COURSE OUTCOMES**

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

- design and execute C programs for stacks, queues, and linked lists.
- apply the different data structures concepts and find solutions to practical problems.
- gain in depth knowledge in searching and sorting programs.
- demonstrate array implementation of list ADT.

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	3	3	-	-	-	-	-	-	-	3	3	3
CO2	3	2	3	3	-	-	-	-	-	-	-	2	3	3
CO3	2	3	1	-	-	-	1	-	-	-	-	2	3	3
CO4	3	3	3	3	-	-	-	-	-	-	-	2	1	3



SEMESTER IV

MA16401

PROBABILITY AND QUEUEING THEORY

3 2 0 4

COURSE OBJECTIVES

- To acquire knowledge of the random variables and manipulate.
- To understand the concepts of standard distributions methods.
- To analyse the relationship between the two random variables.
- To provide necessary basic concepts in probability and random processes related to communication engineering domain.
- To use various queuing theory models for real time situations.

UNIT I RANDOM VARIABLES 15

Axioms of probability – Conditional probability – Total probability – Baye's theorem - Random variable- Probability mass function – Probability density function – Properties – Moments – Moment generating functions and their properties.

UNIT II STANDARD DISTRIBUTION 15

Binomial, Poisson, Geometric, Uniform, Exponential and Normal distributions and their properties – Functions of a random variable.

UNIT III TWO DIMENSIONAL RANDOM VARIABLES 15

Joint distributions – Marginal and conditional distributions – Covariance – Correlation and Linear regression – Transformation of random variables.

UNIT IV RANDOM PROCESS AND MARKOV CHAIN 15

Classification – Stationary process – Poisson process – Markov Chain – Transition probabilities – Limiting Distributions.

UNIT V QUEUEING MODELS 15

Markovian models – (M/M/1), (M/M/C), finite and infinite capacity – (M/G/1) queue – Pollaczek – Khintchine Formula.

TOTAL PERIODS 75

COURSE OUTCOMES

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

- understand the basic probability concepts.
- know the standard distribution for real time applications.
- acquire skills in handling situations involving more than one random variable and functions of random variables.
- evolve with respect to time in a probabilistic manner.
- acquire the fundamental skills to analyze queuing models and systems.

TEXT BOOKS

1. Gross, Donald Harris and M Carl, “ Fundamentals of Queuing Theory”, 3rd ed., Wiley Publications, New Delhi, 2008

2. Ibe. O.C., “Fundamentals of Applied Probability and Random Processes”, Elsevier, 2nd Indian Reprint, 2010.
3. T Veerarajan, “Probability, Statistics and Random Processes”, 2nd ed., Tata McGraw- Hill, New Delhi, 2008.

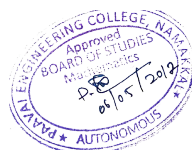
REFERENCES

1. Trivedi, K.S., “Probability and Statistics with Reliability, Queueing and Computer Science Applications”, PHI, New Delhi, 2nd Edition, 2009.
2. Hwei Hsu, “Schaum’s Outline of Theory and Problems of Probability, Random Variables and Random Processes”, Tata McGraw Hill, New Delhi, 9th Reprint, 2010.
3. Yates. R.D. and Goodman. D. J., “Probability and Stochastic Processes”, Wiley India Pvt.Ltd.Bangalore, 2nd Edition, 2012
4. Venkatachalam. G, “Probability and Queueing Theory”, Hitech Publishing Company Pvt.Ltd.,Chennai,3rd Edition, 2012.

WEB LINKS

1. <https://www.youtube.com/watch?v=IYdiKeQ9xEI>
2. <https://www.youtube.com/watch?v=xGkpXk-AnWU>
3. <https://www.youtube.com/watch?v=l-rRtmNpdKU>
4. https://www.youtube.com/watch?v=J70dP_AECzQ
5. <http://172.16.100.200/NPTEL/displayvideo.html?type1=111105041%2Fmod01lec16.mp4>

Mapping of Course Outcomes with Programme Outcomes (1/2/3 indicates strength of correlation) 3-strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	3	-	-	-	-	-	-	-	1	3	2
CO2	3	3	2	3	-	-	-	-	-	-	-	1	3	2
CO3	3	3	2	3	-	-	-	-	-	-	-	1	3	2
CO4	3	3	2	3	-	-	-	-	-	-	-	1	3	2
CO5	3	3	3	2	-	-	-	-	-	-	-	1	3	2



COURSE OBJECTIVES

- To understand the phases in a software project
- To understand fundamental concepts of requirements engineering and analysis modeling.
- To understand the major considerations for enterprise integration and deployment.
- To learn various testing and maintenance measures
- To understand fundamental concepts of requirements engineering and analysis modeling.

UNIT I SOFTWARE PROCESS

9

The Evolving role of Software – Software – **The changing Nature of Software** – Legacy Software – **A generic view of process** – A layered Technology – **A Process Framework** – The Capability Maturity Model Integration(CMMI) – **Process Assessment** – Personal and Team Process Models. Product and Process. Process Models The Waterfall Model –**Incremental Process Models**– Incremental Model – The RAD Model – **Evolutionary Process Models** – Prototyping – **The Spiral Model** – The Concurrent Development Model– **Specialized Process Models** – the Unified Process - **Agile Development**.

UNIT II SOFTWARE REQUIREMENTS

9

Software Engineering Practice – **communication Practice** – Planning practice modeling practice–Construction Practice –**Deployment Requirements Engineering** - **Requirements Engineering tasks** – Initiating the requirements Engineering Process–**Eliciting Requirements** – Developing Use cases – **Building the Analysis Models** - Elements of the Analysis Model – **Analysis pattern** – Negotiating Requirements – **Validating Requirements**.

UNIT III REQUIREMENTS ANALYSIS

9

Requirements Analysis – **Analysis Modeling approaches** – data modeling concepts – **Object oriented Analysis**– Scenario based modeling – **Flow oriented Modeling** – Class based modeling – **creating a behavior model**.

UNIT IV TESTING TECHNIQUES

9

A strategic Approach for Software testing – **Test Strategies for conventional software** – Validation Testing–System Testing – **The Art of Debugging**. **Testing Conventional Applications: Software testing Fundamentals** –Internal and External Views Testing – **White Box Testing** – Basis Path Testing – **Control Structure Testing** – Black Box Testing – **Model Based testing** – Testing for Specialized Environments – **Architectures and Applications** –Patterns for Software Testing.

UNIT V SOFTWARE PROJECT MANAGEMENT

9

Software Cost Estimation – productivity – **Estimation Techniques** – Algorithmic Cost Modeling – **Project Duration and Staffing** - Process and Product Quality – **Quality Assurance and Standards** –Planning – Control- Software Measurement and Metrics - **Process Improvement** – Process Classification – **Measurement** –Analysis and Modeling –Change – **The CMMI process improvement Framework** - Configuration Management. –**Planning Change Management** – Version and Release Management – **System Building** – CASE tools for configuration management.

TOTAL PERIODS 45

COURSE OUTCOMES

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

- identify the key activities in managing a software project.
- compare different process models.
- understand the Concepts of requirements engineering and Analysis Modeling.
- compare and contrast the various testing and maintenance.
- understand the concept of Software Project Management

TEXT BOOKS

1. Roger S. Pressman Software Engineering: A Practitioner's Approach, McGraw Hill International edition, Eighth edition, 2015.
2. Ian Sommerville, Software Engineering, 9 th Edition, Pearson Education, 2011.
3. Watts S. Humphrey, "A Discipline for Software Engineering", Pearson Education, 2007.

REFERENCES

1. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Learning Private Limited, 2009.
2. Pankaj Jalote, "Software Engineering, A Precise Approach", Wiley India, 2010.
3. Kelkar S.A., "Software Engineering", Prentice Hall of India Pvt Ltd, 2007.
4. Stephen R. Schach, "Software Engineering", Tata McGraw-Hill Publishing Company Limited, 2007.
5. James F. Peters and Witold Pedrycz, "Software Engineering, An Engineering Approach", Wiley-India, 2007.

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2
CO5	2	2	2	2	2	-	-	-	-	-	2	1	2	2



COURSE OBJECTIVES

- To understand the relationship between system software and machine architecture
- To know the design and implementation of assemblers.
- To understand the major concept of loader and linker.
- To have an understanding of macro processors.
- To understand the major concept of interactive debugging systems and software tools.

UNIT I INTRODUCTION**8**

System software and machine architecture – The Simplified Instructional Computer (SIC) - Machine architecture- Data and instruction formats - addressing modes -instruction sets - I/O and programming.

UNIT II ASSEMBLERS**10**

Basic assembler functions - A simple SIC assembler – Assembler algorithm and data structures - Machine dependent assembler features- Instruction formats and addressing modes – Program relocation - Machine independent assembler features - Literals-Symbol-defining statements – Expressions - One pass assemblers and Multi pass assemblers - Implementation example - MASM assembler.

UNIT III LOADERS AND LINKERS**9**

Basic loader functions - Design of an Absolute Loader – A Simple Bootstrap Loader -Machine dependent loader features - Relocation – Program Linking – Algorithm and Data Structures for Linking Loader -Machine- independent loader features –Automatic Library Search – Loader Options - Loader design options - Linkage Editors – Dynamic Linking – Bootstrap Loaders - Implementation example - MSDOS linker.

UNIT IV MACRO PROCESSORS**9**

Basic macro processor functions - Macro Definition and Expansion – Macro Processor Algorithm and data structures - Machine-independent macro processor features -Concatenation of Macro Parameters – Generation of Unique Labels – Conditional Macro Expansion – Keyword Macro Parameters-Macro within Macro- Implementation example -MASM Macro Processor – ANSI C Macro language.

UNIT V SYSTEM SOFTWARE TOOLS**9**

Text editors - Overview of the Editing Process - User Interface – Editor Structure. Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User-Interface Criteria.

TOTAL PERIODS 45**COURSE OUTCOMES**

At the end if this course, students will be able to

- identify the approach of different machine architecture.
- study of machine dependent and independent assembler algorithms and program relocation.
- design of various linker loader and program linking.
- study of machine independent macro processors.
- understand the text editors and debugging systems.

TEXT BOOKS

1. Leland L. Beck, “System Software – An Introduction to Systems Programming”, 3rd Edition, Pearson Education Asia, 2006.

REFERENCES

1. D. M. Dhamdhare, “Systems Programming and Operating Systems”, Second Revised Edition, Tata McGraw-Hill 2000.
2. John J. Donovan “Systems Programming”, Tata McGraw-Hill Edition, 2000.
3. John R. Levine, Linkers & Loaders – Harcourt India Pvt. Ltd., Morgan Kaufmann Publishers, 2000
company, 1994

WEB LINKS

1. [.http://study.com/academy/topic/systems-software.html](http://study.com/academy/topic/systems-software.html)
2. https://www.youtube.com/watch?v=VG9VopzV_T0
3. https://www.youtube.com/watch?v=6ipFf3vLifU&list=PLRjiB7KcljoS22wmROkUKZ8zD4_Fj8U2R

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2
CO5	2	2	2	2	2	-	-	-	-	-	2	1	2	2



COURSE OBJECTIVES

- To explore the fundamentals of database management systems.
- To make the students understand the relational model.
- To familiarize database design.
- To familiarize with the different types of transaction concepts.
- To make the students understand the implementation and security issues in databases.

UNIT 1 INTRODUCTION**9**

Purpose of Database System -- Views of data – Data Models – Database Languages – Database System Architecture – Database users and Administrator – Entity-Relationship model (E-R model) – E-RDiagrams -- Introduction to relational databases.

UNIT II RELATIONAL MODEL**9**

The relational Model – The catalog- Types– Keys - Relational Algebra – Domain Relational Calculus – Tuple Relational Calculus - Fundamental operations – Additional I/O operations- SQL fundamentals -Integrity –Triggers - Security – Advanced SQL features –Embedded SQL– Dynamic SQL- Missing Information– Views– Introduction to Distributed Databases and Client/Server Databases.

UNIT III DATABASE DESIGN**9**

Functional Dependencies – Non-loss Decomposition – Functional Dependencies – First, Second, Third Normal Forms, Dependency Preservation – Boyce/ Code Normal Form-Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form.

UNIT IV TRANSACTIONS**9**

Transaction Concepts - Transaction Recovery – ACID Properties– System Recovery –Media Recovery –Two Phase Commit - Save Points – SQL Facilities for recovery –Concurrency – Need for Concurrency –Locking Protocols – Two Phase Locking –Intent Locking – Deadlock- Serializability – Recovery solution Levels –ISQL Facilities for Concurrency.

UNIT V IMPLEMENTATION TECHNIQUES**9**

Overview of Physical Storage Media –Magnetic Disks – RAID – Tertiary storage –File Organization – Organization of Records in Files – Indexing and Hashing –Ordered Indices – B+ tree Index Files -B tree-Index Files – Static Hashing – Dynamic Hashing –Query Processing Overview – Catalog Information for Cost Estimation- Selection Operation – Sorting –Join Operation – Database Tuning.

TOTAL PERIODS 45**COURSE OUTCOMES**

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

- use the relational model, ER diagrams.
- write queries in structural query language.
- design the database using various normal forms.
- understand the transaction concepts and locking protocols.

- implement database concepts with security.

TEXT BOOKS

1. Silberschatz, H.Korth and Sudarshan S., “Database System Concepts”, 6th Edition, McGraw-Hill International, 2010.

REFERENCES

1. Elmasri R. and Shamkant B. Navathe, “Fundamentals of Database Systems”, 6th Edition, AddisonWesley, 2011.
2. AtulKahate, “Introduction to Database Management Systems”, Pearson Education, New Delhi, 2006.
3. Raghu Ramakrishnan, “Database Management Systems”, Fourth Edition, Tata McGraw Hill, 2010.
4. G.K.Gupta, “Database Management Systems”, Tata McGraw Hill, 2011.
5. C.J.Date, A.Kannan, S.Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

WEB LINKS

1. <http://www.nptelvideos.in/2012/11/database-management-system.html>
2. <https://www.youtube.com/watch?v=1057YmExS-I>
3. <http://freevidelectures.com/Course/2668/Database-Management-System>

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2
CO5	2	2	2	2	2	-	-	-	-	-	2	1	2	2



COURSE OBJECTIVES

- To understand the concepts of data communications
- To be familiar with the Transmission media and Tools
- To study the functions of OSI layers
- To learn about IEEE standards in computer networking
- To get familiarized with different protocols and network components.

UNIT I FUNDAMENTALS AND LINK LAYER 9

Building a network – Requirements – Layering and protocols – Internet Architecture – Network software – Performance ; Link layer Services – Framing – Error Detection – Flow control

UNIT II MEDIA ACCESS AND INTERNET WORKING 9

Media access control – Ethernet (802.3) – Wireless LAN' s – 802.11 – Bluetooth – Switching and bridging – Basic Internetworking (IP, CIDR, ARP, DHCP, ICMP)

UNIT III ROUTING 9

Routing (RIP, OSPF, metrics) – Switch basics – Global Internet (Areas, BGP, IPv6), Multicast – addresses – multicast routing (DVMRP, PIM, MSDB, MPLS) – Routing among Mobile Devices.

UNIT IV TRANSPORT LAYER 9

Overview of Transport layer – UDP – Reliable byte stream (TCP) – Connection management – Flow control – Retransmission – TCP Congestion control – Congestion avoidance (DECbit, RED) – QoS – Application requirements

UNIT V APPLICATION LAYER 9

Traditional applications -Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – Web Services – DNS –SNMP – Overlay networks.

TOTAL PERIODS 45

COURSE OUTCOMES

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

- identify the components required to build different types of networks.
- choose the required functionality at each layer for given application.
- identify solution for each functionality at each layer.
- trace the flow of information from one node to another node in the network.
- understanding the Applications of Networks and data communications.

TEXT BOOKS

1. Behrouz A. Forouzan, “Data communication and Networking”, Fourth Edition, Tata McGraw – Hill, 2011
2. Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Fifth Edition, Morgan Kaufmann Publishers, 2011.

REFERENCES

1. Andrew S. Tanenbaum, Computer Networks, Pearson Education, 2008
2. James F. Kurose, Keith W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", Fifth Edition, Pearson Education, 2009.
3. Nader. F. Mir, "Computer and Communication Networks", Pearson Prentice Hall Publishers, 2010.
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw Hill Publisher, 2011.
5. William Stallings, "Data and Computer Communication", Sixth Edition, Pearson Education, 2000

WEB LINKS

1. <https://www.youtube.com/watch?v=3DZLIItfbqtQ&list=PL1EC310A0BF4B2CA7>
2. <https://www.youtube.com/watch?v=zzXs0EnCin0>
3. <https://www.youtube.com/watch?v=aNqiTCZ-nko>

Mapping of Course Outcomes with Programming Outcomes (1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2
CO5	2	2	2	2	2	-	-	-	-	-	2	1	2	2



COURSE OBJECTIVES

- To create and use a database
- To have hands on experience on DDL Commands
- To have a good understanding of DML Commands and DCL commands
- To be familiarize with a query language

LIST OF EXPERIMENTS

1. Data Definition, Table Creation, Constraints.
2. Insert, Select Commands, Update & Delete Commands.
3. Nested Queries & Join Queries
4. Views
5. High level programming language extensions (Control structures, Procedures and Functions).
6. Front end tools
7. Forms
8. Triggers
9. Menu Design
10. Reports.
11. Database Design and implementation (Mini Project).
 - a. Personal Information System.
 - b. Web Based User Identification System.
 - c. Timetable Management System.
 - d. Hotel Management System

**TOTAL PERIODS 60****COURSE OUTCOMES**

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

- design and implement a database schema for a given problem-domain
- populate and query a database
- create and maintain tables using PL/SQL
- prepare reports

Mapping of Course Outcomes with Programming Outcomes
(1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak

COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2

COURSE OBJECTIVES

- To learn socket programming.
- To be familiar with simulation tools.
- performance of the protocols in different layers
- To have hands on experience on various networking protocols.

LIST OF EXPERIMENTS

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client – Server model
3. Write a code simulating ARP /RARP protocols.
4. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.
 - Link State routing
 - Distance vector
5. Applications using TCP Sockets like
 - Echo client and echo server
 - File transfer
 - Remote command execution
 - 5.4 Chat
6. Applications using TCP and UDP Sockets like
 - DNS
7. Applications using Raw Sockets like
 - Ping
 - 7.2. Trace route
8. Write a program to implement RPC (Remote Procedure Call)
9. Study of Network simulator (NS).and Simulation of Congestion Control Algorithms using NS
10. Study of TCP/UDP performance

TOTAL PERIODS 60**COURSE OUTCOMES**

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

- use simulation tools
- implement the various protocols.
- analyze the performance of the protocols in different layers.
- analyze various routing algorithms

REFERENCE: Spoken-tutorial.org.

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**SOFTWARE**

- C / C++ / Java / Equivalent Compiler 30
- Network simulator like NS2/Glomosim/OPNET/ Equivalent

HARDWARE: Standalone desktops

Mapping of Course Outcomes with Programming Outcomes
(1/2/3 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak

COs	Programme Outcomes(POs)												Programme Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO 2
CO1	1	1	1	1	1	-	-	-	-	-	1	1	1	1
CO2	2	2	2	2	2	-	-	-	-	-	1	1	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2	3	3
CO4	2	2	2	2	2	-	-	-	-	-	2	1	2	2



COURSE OBJECTIVES

enable the students to

- Introduce students to ideas and techniques from discrete mathematics that are widely used in science and engineering.
- Make the students to think logically and mathematically and apply these techniques in solving problems.
- Provide the foundation for imbedding logical reasoning in computer science.
- Develop recursive algorithms based on mathematical induction.
- Know basic properties of relations.

UNIT I PROPOSITIONAL CALCULUS

9

Propositions - Logical connectives - Compound propositions - Conditional and conditional propositions - Truth tables - Tautologies and contradictions - Contrapositive - Logical equivalences and implications - DeMorgan's Laws - Normal forms - Principal conjunctive and disjunctive normal forms - Rules of inference - Arguments - Validity of arguments.

UNIT II PREDICATE CALCULUS

9

Predicates - Statement function - Variables - Free and bound variables - Quantifiers - Universe of discourse - Logical equivalences and implications for quantified statements - Theory of inference - The rules of universal specification and generalization - Validity of arguments.

UNIT-III SET THEORY

9

Basic concepts - Notations - Subset - Algebra of sets - The power set - Ordered pairs and Cartesian product - Relations on sets - Types of relations and their properties - Relational matrix and the graph of relation - Partitions Equivalence relations.

UNIT IV FUNCTIONS

9

Definitions of functions - Classification of functions - Type of functions - Examples - Composition of functions - Inverse functions - Binary and n - ary operations - Characteristic function of a set - Hashing functions - Recursive functions - Permutation functions.

UNIT-V LATTICE THEORY

9

Partial ordering - Posets - Lattices as Posets - Properties of lattices - Lattices as Algebraic systems - Sub lattices - Direct product and Homomorphism - Some Special lattices.

TOTAL PERIODS 45**COURSE OUTCOMES**

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

- Construct mathematical arguments using logical connectives and quantifiers.
- Verify the correctness of an argument using propositional and predicate logic and truth tables.
- Demonstrate the ability to solve problems using counting techniques and combinatorics Construct proofs using direct proof, proof by contraposition, proof by contradiction, and proof by cases.
- Perform operations on discrete structures such as sets, functions, relations, and sequences.
- Understand the concepts of Boolean algebra.

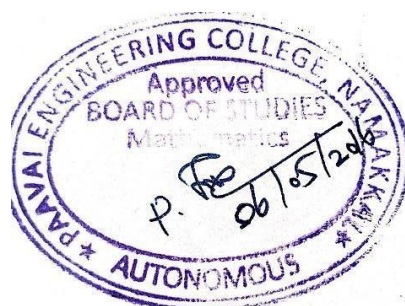
TEXT BOOKS

1. Kenneth H.Rosen, "Discrete Mathematics and its Applications (with Combinatorics and Graph Theory)", 6th Edition, Tata McGraw - Hill, 5th Reprint 2008.
2. Trembly J.P and Manohar.R, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw - Hill, 35th Reprint 2008.

REFERENCES

1. Ralph.P.Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th Edition, Pearson Education, 2002.
2. A.Tamilarasi, A.M.Natarajan, "Discrete Mathematics and its Applications", 3rd Edition, Khanna Publishers, 2008.
3. T.Veerarajan, "Discrete Mathematics with Graph Theory and Combinatorics", Tata McGraw - Hill, 2007.

Mapping of Course Outcomes with Programme Outcomes (1/2/3 indicates strength of correlation) 3-strong, 2-Medium, 1-Weak														
COs	Programme Outcomes(POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO2	3	2	3	3	-	-	-	-	-	-	-	2	3	3
CO3	2	3	3	-	-	-	-	-	-	-	-	3	3	3
CO4	2	2	3	3	-	-	-	-	-	-	-	2	3	3
CO5	3	3	3	-	-	-	-	-	-	-	-	3	3	3



OBJECTIVES:

The student should be made to:

- Learn Java Programming.
- Understand different Internet Technologies.
- Be exposed to java specific web services architecture.

UNIT I JAVA PROGRAMMING**9**

An overview of Java – Data Types – Variables and Arrays – Operators – Control Statements – Classes – Objects – Methods – Inheritance – Packages – Abstract classes – Interfaces and Inner classes – Exception handling – Introduction to Threads – Multithreading – String handling – Streams and I/O – Applets.

UNIT II WEBSITES BASICS, HTML 5, CSS 3, WEB 2.0**8**

Web 2.0: Basics-RIA Rich Internet Applications - Collaborations tools - **Understanding websites and web servers:** Understanding Internet – Difference between websites and web server- Internet technologies Overview – Understanding the difference between internet and intranet; **HTML and CSS:** HTML 5.0 , XHTML, CSS 3.

UNIT III CLIENT SIDE AND SERVER SIDE PROGRAMMING**11**

Java Script: An introduction to JavaScript-JavaScript DOM Model-Date and Objects,-Regular Expressions- Exception Handling-Validation-Built-in objects-Event Handling- DHTML with JavaScript. **Servlets:** Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies- Installing and Configuring Apache Tomcat Web Server;- **DATABASE CONNECTIVITY:** JDBC perspectives, JDBC program example - **JSP:** Understanding Java Server Pages-JSP Standard Tag Library(JSTL)-Creating HTML forms by embedding JSP code.

UNIT IV PHP and XML**8**

An introduction to PHP: PHP- Using PHP- Variables- Program control- Built-in functions-Connecting to Database – Using Cookies-Regular Expressions; **XML:** Basic XML- Document Type Definition- XML Schema DOM and Presenting XML, XML Parsers and Validation, XSL and XSLT Transformation, News Feed (RSS and ATOM).

UNIT V INTRODUCTION TO AJAX and WEB SERVICES**9**

AJAX: Ajax Client Server Architecture-XML Http Request Object-Call Back Methods; **Web Services:** Introduction- Java web services Basics – Creating, Publishing ,Testing and Describing a Web services (WSDL)-Consuming a web service, Database Driven web service from an application – SOAP.

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

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

- Implement Java programs.
- Create a basic website using HTML and Cascading Style Sheets.
- Design and implement dynamic web page with validation using JavaScript objects and by applying different event handling mechanisms.
- Design rich client presentation using AJAX.
- Design and implement simple web page in PHP, and to present data in XML format.
- Design and implement server side programs using Servlets and JSP.

TEXT BOOKS:

1. Deitel and Deitel and Nieto, "Internet and World Wide Web - How to Program", Prentice Hall, 5th Edition, 2011.
2. Herbert Schildt, "Java-The Complete Reference", Eighth Edition, Mc Graw Hill Professional, 2011.

REFERENCES:

1. Stephen Wynkoop and John Burke "Running a Perfect Website", QUE, 2nd Edition, 1999.
2. Chris Bates, Web Programming – Building Intranet Applications, 3rd Edition, Wiley Publications, 2009.
3. Jeffrey C and Jackson, "Web Technologies A Computer Science Perspective", Pearson Education, 2011.
4. Gopalan N.P. and Akilandeswari J., "Web Technology", Prentice Hall of India, 2011.
5. Paul Dietel and Harvey Deitel, "Java How to Program", , 8th Edition Prentice Hall of India.
6. Mahesh P. Matha, "Core Java A Comprehensive Study", Prentice Hall of India, 2011.
7. Uttam K.Roy, "Web Technologies", Oxford University Press, 2011.

CS6502**OBJECT ORIENTED ANALYSIS AND DESIGN**
L T P C
3 0 0 3
OBJECTIVES:**The student should be made to:**

- Learn the basics of OO analysis and design skills.
- Learn the UML design diagrams.
- Learn to map design to code.
- Be exposed to the various testing techniques.

UNIT I UML DIAGRAMS**9**

Introduction to OOAD – Unified Process - UML diagrams – Use Case – Class Diagrams– Interaction Diagrams – State Diagrams – Activity Diagrams – Package, component and Deployment Diagrams.

UNIT II DESIGN PATTERNS**9**

GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller - Design Patterns – creational - factory method - structural – Bridge – Adapter - behavioral – Strategy – observer.

UNIT III CASE STUDY**9**

Case study – the Next Gen POS system, Inception -Use case Modeling - Relating Use cases – include, extend and generalization - Elaboration - Domain Models - Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies - Aggregation and Composition.

UNIT IV APPLYING DESIGN PATTERNS**9**

System sequence diagrams - Relationship between sequence diagrams and use cases Logical architecture and UML package diagram – Logical architecture refinement - UML class diagrams - UML interaction diagrams - Applying GoF design patterns.

UNIT V CODING AND TESTING**9**

Mapping design to code – Testing: Issues in OO Testing – Class Testing – OO Integration Testing – GUI Testing – OO System Testing.

TOTAL: 45 PERIODS

UNIT IV TURING MACHINES

9

Definitions of Turing machines – Models – Computable languages and functions – Techniques for Turing machine construction – Multi head and Multi tape Turing Machines - The Halting problem – Partial Solvability – Problems about Turing machine- Chomskian hierarchy of languages.

UNIT V UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS

9

Unsolvability Problems and Computable Functions – Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. MEASURING AND CLASSIFYING COMPLEXITY: Tractable and Intractable problems- Tractable and possibly intractable problems - P and NP completeness - Polynomial time reductions.

TOTAL: 45 PERIODS

OUTCOMES:

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

- Design Finite State Machine, Pushdown Automata, and Turing Machine.
- Explain the Decidability or Undecidability of various problems

TEXT BOOKS:

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3)
2. John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (UNIT 4,5)

REFERENCES:

1. Mishra K L P and Chandrasekaran N, "Theory of Computer Science - Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.
2. Harry R Lewis and Christos H Papadimitriou, "Elements of the Theory of Computation", Second Edition, Prentice Hall of India, Pearson Education, New Delhi, 2003.
3. Peter Linz, "An Introduction to Formal Language and Automata", Third Edition, Narosa Publishers, New Delhi, 2002.
4. Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009

CS6504

COMPUTER GRAPHICS

L T P C
3 0 0 3

OBJECTIVES:

The student should be made to:

- Gain knowledge about graphics hardware devices and software used.
- Understand the two dimensional graphics and their transformations.
- Understand the three dimensional graphics and their transformations.
- Appreciate illumination and color models.
- Be familiar with understand clipping techniques.

UNIT I INTRODUCTION

9

Survey of computer graphics, Overview of graphics systems – Video display devices, Raster scan systems, Random scan systems, Graphics monitors and Workstations, Input devices, Hard copy Devices, Graphics Software; Output primitives – points and lines, line drawing algorithms, loading the frame buffer, line function; circle and ellipse generating algorithms; Pixel addressing and object geometry, filled area primitives.

UNIT II TWO DIMENSIONAL GRAPHICS

9

Two dimensional geometric transformations – Matrix representations and homogeneous coordinates, composite transformations; Two dimensional viewing – viewing pipeline, viewing coordinate reference frame; window-to-viewport coordinate transformation, Two dimensional viewing functions; clipping operations – point, line, and polygon clipping algorithms.

UNIT III THREE DIMENSIONAL GRAPHICS

10

Three dimensional concepts; Three dimensional object representations – Polygon surfaces- Polygon tables- Plane equations - Polygon meshes; Curved Lines and surfaces, Quadratic surfaces; Blobby objects; Spline representations – Bezier curves and surfaces -B-Spline curves and surfaces. TRANSFORMATION AND VIEWING: Three dimensional geometric and modeling transformations – Translation, Rotation, Scaling, composite transformations; Three dimensional viewing – viewing pipeline, viewing coordinates, Projections, Clipping; Visible surface detection methods.

UNIT IV ILLUMINATION AND COLOUR MODELS

7

Light sources - basic illumination models – halftone patterns and dithering techniques; Properties of light - Standard primaries and chromaticity diagram; Intuitive colour concepts - RGB colour model - YIQ colour model - CMY colour model - HSV colour model - HLS colour model; Colour selection.

UNIT V ANIMATIONS & REALISM

10

ANIMATION GRAPHICS: Design of Animation sequences – animation function – raster animation – key frame systems – motion specification –morphing – tweening. COMPUTER GRAPHICS REALISM: Tiling the plane – Recursively defined curves – Koch curves – C curves – Dragons – space filling curves – fractals – Grammar based models – fractals – turtle graphics – ray tracing.

TOTAL: 45 PERIODS

OUTCOMES:

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

- Design two dimensional graphics.
- Apply two dimensional transformations.
- Design three dimensional graphics.
- Apply three dimensional transformations.
- Apply Illumination and color models.
- Apply clipping techniques to graphics.
- Design animation sequences.

TEXT BOOKS:

1. John F. Hughes, Andries Van Dam, Morgan Mc Guire ,David F. Sklar , James D. Foley, Steven K. Feiner and Kurt Akeley ,”Computer Graphics: Principles and Practice”, , 3rd Edition, Addison-Wesley Professional,2013. (UNIT I, II, III, IV).
2. Donald Hearn and Pauline Baker M, “Computer Graphics”, Prentice Hall, New Delhi, 2007 (UNIT V).

REFERENCES:

1. Donald Hearn and M. Pauline Baker, Warren Carithers,“Computer Graphics With Open GL”, 4th Edition, Pearson Education, 2010.
2. Jeffrey McConnell, “Computer Graphics: Theory into Practice”, Jones and Bartlett Publishers, 2006.
3. Hill F S Jr., "Computer Graphics", Maxwell Macmillan” , 1990.
4. Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen R Marschner, Erik Reinhard, Kelvin Sung, and AK Peters, Fundamental of Computer Graphics, CRC Press, 2010.
5. William M. Newman and Robert F.Spruill, “Principles of Interactive Computer Graphics”, Mc Graw Hill 1978.
6. <http://nptel.ac.in/>

OBJECTIVES:

The student should be made to:

- Learn the basics of OO analysis and design skills.
- Be exposed to the UML design diagrams.
- Learn to map design to code.
- Be familiar with the various testing techniques

LIST OF EXPERIMENTS:

To develop a mini-project by following the 9 exercises listed below.

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
7. Develop and test the Technical services layer.
8. Develop and test the Domain objects layer.
9. Develop and test the User interface layer.

SUGGESTED DOMAINS FOR MINI-PROJECT:

1. Passport automation system.
2. Book bank
3. Exam Registration
4. Stock maintenance system.
5. Online course reservation system
6. E-ticketing
7. Software personnel management system
8. Credit card processing
9. e-book management system
10. Recruitment system
11. Foreign trading system
12. Conference Management System
13. BPO Management System
14. Library Management System
15. Student Information System

TOTAL: 45 PERIODS

OUTCOMES:

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

- Design and implement projects using OO concepts.
- Use the UML analysis and design diagrams.
- Apply appropriate design patterns.
- Create code from design.
- Compare and contrast various testing techniques

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

Suggested Software Tools:

Rational Suite (or) Argo UML (or) equivalent, Eclipse IDE and JUnit

Software Tools

30 user License

Rational Suite

Open Source Alternatives: ArgoUML, Visual

Paradigm

Eclipse IDE and JUnit

PCs

30

CS6512

INTERNET PROGRAMMING LABORATORY

L T P C
0 0 3 2

OBJECTIVES:

The student should be made to:

- Be familiar with Web page design using HTML/XML and style sheets
- Be exposed to creation of user interfaces using Java frames and applets.
- Learn to create dynamic web pages using server side scripting.
- Learn to write Client Server applications.
- Be familiar with the frameworks JSP Strut, Hibernate, Spring
- Be exposed to creating applications with AJAX

LIST OF EXPERIMENTS:

IMPLEMENT THE FOLLOWING:

WEBPAGE CONCEPTS

- a) Create a web page with the following using HTML
 - a. To embed a map in a web page
 - b. To fix the hot spots in that map
 - c. Show all the related information when the hot spots are clicked.
- b) Create a web page with the following.
 - a. Cascading style sheets.
 - b. Embedded style sheets.
 - c. Inline style sheets. Use our college information for the web pages.
- c) Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document.

SOCKETS & SERVLETS

- a) Write programs in Java using sockets to implement the following:
 - i. HTTP request
 - ii. FTP
 - iii. SMTP
 - iv. POP3
- b) Write a program in Java for creating simple chat application with datagram sockets and datagram packets.
- c) Write programs in Java using Servlets:
 - i. To invoke servlets from HTML forms

- ii. To invoke servlets from Applets
- d) Write programs in Java to create three-tier applications using servlets for conducting on-line examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.
- e) Write a program to lock servlet itself to a particular server IP address and port number. It requires an init parameter key that is appropriate for its servlet IP address and port before it unlocks itself and handles a request
- f) Session tracking using hidden form fields and Session tracking for a hit count
- g) Install TOMCAT web server. Convert the static webpages of programs 1&2 into dynamic web pages using servlets (or JSP) and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

ADVANCE CONCEPTS:

- a) Implement a simple program using following frameworks
 - a. JSP Struts Framework b. Hibernate c. Spring
- b) Explore the following application in AJAX: Searching in real time with live searches, Getting the answer with auto complete, Chatting with friends ,Dragging and dropping with Ajax, Getting instant login feedback, Ajax-enabled popup menus, Modifying Web pages on the fly.
- c) Write a web services for finding what people think by asking 500 people's opinion for any consumer product
- d) Write a web services for predicting for any product sales

TOTAL: 45 PERIODS

OUTCOMES:

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

- Design Web pages using HTML/XML and style sheets
- Create user interfaces using Java frames and applets.
- Create dynamic web pages using server side scripting.
- Write Client Server applications.
- Use the frameworks JSP Strut, Hibernate, Spring
- Create applications with AJAX

REFERENCE:

spoken-tutorial.org.

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

SOFTWARE:

Java, Dream Weaver or Equivalent, MySQL or Equivalent, Apache Server

HARDWARE:

Standalone desktops 30 Nos

CS6513

COMPUTER GRAPHICS LABORATORY

L T P C

0 0 3 2

OBJECTIVES:

The student should be made to:

- Understand graphics programming
- Be exposed to creation of 3D graphical scenes using open graphics library suits
- Be familiar with image manipulation, enhancement
- Learn to create animations
- To create a multimedia presentation/Game/Project.

LIST OF EXPERIMENTS:

IMPLEMENT THE EXERCISES USING C / OPENGL / JAVA

1. Implementation of Algorithms for drawing 2D Primitives – Line (DDA, Bresenham) – all slopes
Circle (Midpoint)
2. 2D Geometric transformations –
Translation
Rotation Scaling
Reflection Shear
Window-Viewport
3. Composite 2D Transformations
4. Line Clipping
5. 3D Transformations - Translation, Rotation, Scaling.
6. 3D Projections – Parallel, Perspective.
7. Creating 3D Scenes.
8. Image Editing and Manipulation - Basic Operations on image using any image editing software, Creating gif animated images, Image optimization.
9. 2D Animation – To create Interactive animation using any authoring tool.

TOTAL: 45 PERIODS

OUTCOMES:

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

- Create 3D graphical scenes using open graphics library suits
- Implement image manipulation and enhancement
- Create 2D animations using tools

REFERENCE:

spoken-tutorial.org

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

SOFTWARE

C, C++, Java, OpenGL

HARDWARE:

Standalone desktops - 30 Nos.
(or)
Server supporting 30 terminals or more.

CS6601

DISTRIBUTED SYSTEMS

L T P C
3 0 0 3

OBJECTIVES:

The student should be made to:

- Understand foundations of Distributed Systems.
- Introduce the idea of peer to peer services and file system.
- Understand in detail the system level and support required for distributed system.
- Understand the issues involved in studying process and resource management.

UNIT III KNOWLEDGE INFERENCE

9

Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network-Dempster - Shafer theory.

UNIT IV PLANNING AND MACHINE LEARNING

9

Basic plan generation systems - Strips -Advanced plan generation systems – K strips -Strategic explanations -Why, Why not and how explanations. Learning- Machine learning, adaptive Learning.

UNIT V EXPERT SYSTEMS

9

Expert systems - Architecture of expert systems, Roles of expert systems - Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems - MYCIN, DART, XOON, Expert systems shells.

TOTAL: 45 PERIODS

OUTCOMES:

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

- Identify problems that are amenable to solution by AI methods.
- Identify appropriate AI methods to solve a given problem.
- Formalise a given problem in the language/framework of different AI methods.
- Implement basic AI algorithms.
- Design and carry out an empirical evaluation of different algorithms on a problem formalisation, and state the conclusions that the evaluation supports.

TEXT BOOKS:

1. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill- 2008. (Units-I,II,VI & V)
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007. (Unit-III).

REFERENCES:

1. Peter Jackson, “Introduction to Expert Systems”, 3rd Edition, Pearson Education, 2007.
2. Stuart Russel and Peter Norvig “AI – A Modern Approach”, 2nd Edition, Pearson Education 2007.
3. Deepak Khemani “Artificial Intelligence”, Tata Mc Graw Hill Education 2013.
4. <http://nptel.ac.in>

CS6611

MOBILE APPLICATION DEVELOPMENT LABORATORY

L T P C
0 0 3 2

OBJECTIVES:

The student should be made to:

- Know the components and structure of mobile application development frameworks for Android and windows OS based mobiles.
- Understand how to work with various mobile application development frameworks.
- Learn the basic and important design concepts and issues of development of mobile applications.
- Understand the capabilities and limitations of mobile devices.

LIST OF EXPERIMENTS:

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.

5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi threading
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Write a mobile application that creates alarm clock

TOTAL: 45 PERIODS

OUTCOMES:

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

- Design and Implement various mobile applications using emulators.
- Deploy applications to hand-held devices

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

Standalone desktops with Windows or Android or

iOS or Equivalent Mobile Application Development

Tools with appropriate emulators and debuggers - 30 Nos.

CS6612

COMPILER LABORATORY

L T P C
0 0 3 2

OBJECTIVES:

The student should be made to:

- Be exposed to compiler writing tools.
- Learn to implement the different Phases of compiler
- Be familiar with control flow and data flow analysis
- Learn simple optimization techniques

LIST OF EXPERIMENTS:

1. Implementation of Symbol Table
2. Develop a lexical analyzer to recognize a few patterns in C.
(Ex. identifiers, constants, comments, operators etc.)
3. Implementation of Lexical Analyzer using Lex Tool
4. Generate YACC specification for a few syntactic categories.
 - a) Program to recognize a valid arithmetic expression that uses operator +, -, *, and /.
 - b) Program to recognize a valid variable which starts with a letter followed by any number of letters or digits.
 - d) Implementation of Calculator using LEX and YACC
5. Convert the BNF rules into Yacc form and write code to generate Abstract Syntax Tree.
6. Implement type checking
7. Implement control flow analysis and Data flow Analysis
8. Implement any one storage allocation strategies (Heap, Stack, Static)
9. Construction of DAG
10. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using a 8086 assembler. The target assembly instructions can be simple move, add, sub, jump. Also simple addressing modes are used.
11. Implementation of Simple Code Optimization Techniques (Constant Folding., etc.)

TOTAL: 45 PERIODS

OUTCOMES:

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

- Implement the different Phases of compiler using tools
- Analyze the control flow and data flow of a typical program
- Optimize a given program
- Generate an assembly language program equivalent to a source language program

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:

Standalone desktops with C / C++ compiler and Compiler writing tools 30 Nos.
(or)

Server with C / C++ compiler and Compiler writing tools supporting 30 terminals or more.

LEX and YACC

GE6674 COMMUNICATION AND SOFT SKILLS- LABORATORY BASED L T P C
0 0 4 2

OBJECTIVES:

To enable learners to,

- Develop their communicative competence in English with specific reference to speaking and listening
- Enhance their ability to communicate effectively in interviews.
- Strengthen their prospects of success in competitive examinations.

UNIT I LISTENING AND SPEAKING SKILLS 12
Conversational skills (formal and informal)- **group discussion**- making effective presentations using computers, listening/watching interviews conversations, documentaries. **Listening to lectures**, discussions from TV/ Radio/ **Podcast**.

UNIT II READING AND WRITING SKILLS 12
Reading different genres of texts ranging from newspapers to creative writing. Writing job applications- **cover letter**- resume- emails- **letters**- memos- reports. **Writing abstracts**- summaries- **interpreting visual texts**.

UNIT III ENGLISH FOR NATIONAL AND INTERNATIONAL EXAMINATIONS AND PLACEMENTS 12
International English Language Testing System (IELTS) - Test of English as a Foreign Language (TOEFL) - **Civil Service(Language related)**- Verbal Ability.

UNIT IV INTERVIEW SKILLS 12
Different types of Interview format- **answering questions**- offering information- mock interviews-body language(paralinguistic features)- **articulation of sounds**- intonation.

MG2453

RESOURCE MANAGEMENT TECHNIQUES

L T P C
3 0 0 3

UNIT I LINEAR PROGRAMMING:

9

Principal components of decision problem – Modeling phases – LP Formulation and graphic solution – Resource allocation problems – Simplex method – Sensitivity analysis.

UNIT II DUALITY AND NETWORKS:

9

Definition of dual problem – Primal – Dual relation ships – Dual simplex methods – Post optimality analysis – Transportation and assignment model shortest route problem.

UNIT III INTEGER PROGRAMMING:

9

Cutting plan algorithm – Branch and bound methods, Multistage (Dynamic) programming.

UNIT IV CLASSICAL OPTIMISATION THEORY:

9

Unconstrained external problems, Newton – Ralphson method – Equality constraints – Jacobean methods – Lagrangian method – Kuhn – Tucker conditions – Simple problems.

UNIT V OBJECT SCHEDULING:

9

Network diagram representation – Critical path method – Time charts and resource leveling – PERT.

TOTAL: 45 PERIODS

REFERENCES:

1. Anderson 'Quantitative Methods for Business', 8th Edition, Thomson Learning, 2002.
2. Winston 'Operation Research', Thomson Learning, 2003.
3. H.A.Taha, 'Operation Research', Prentice Hall of India, 2002.
4. Vohra, 'Quantitative Techniques in Management', Tata McGraw Hill, 2002.
5. Anand Sarma, 'Operation Research', Himalaya Publishing House, 2003.

CS2032

DATA WAREHOUSING AND DATA MINING

L T P C
3 0 0 3

UNIT I DATA WAREHOUSING

10

Data warehousing Components – Building a Data warehouse – Mapping the Data Warehouse to a Multiprocessor Architecture – DBMS Schemas for Decision Support – Data Extraction, Cleanup, and Transformation Tools – Metadata.

UNIT II BUSINESS ANALYSIS

8

Reporting and Query tools and Applications – Tool Categories – The Need for Applications – Cognos Impromptu – Online Analytical Processing (OLAP) – Need – Multidimensional Data Model – OLAP Guidelines – Multidimensional versus Multirelational OLAP – Categories of Tools – OLAP Tools and the Internet.

OBJECTIVES:

The student should be made to:

- Be exposed to the different cipher techniques
- Learn to implement the algorithms DES, RSA, MD5, SHA-1
- Learn to use network security tools like GnuPG, KF sensor, Net Strumbler

LIST OF EXPERIMENTS:

1. Implement the following SUBSTITUTION & TRANSPOSITION TECHNIQUES concepts:
 - a) Caesar Cipher
 - b) Playfair Cipher
 - c) Hill Cipher
 - d) Vigenere Cipher
 - e) Rail fence – row & Column Transformation
2. Implement the following algorithms
 - a) DES
 - b) RSA Algorithm
 - c) Diffie-Hellman
 - d) MD5
 - e) SHA-1
5. Implement the SIGNATURE SCHEME - Digital Signature Standard
6. Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG).
7. Setup a honey pot and monitor the honeypot on network (KF Sensor)
8. Installation of rootkits and study about the variety of options
9. Perform wireless audit on an access point or a router and decrypt WEP and WPA. (Net Stumbler)
10. Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w)

TOTAL: 45 PERIODS

OUTCOMES:

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

- Implement the cipher techniques
- Develop the various security algorithms
- Use different open source tools for network security and analysis

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:**SOFTWARE:**

C / C++ / Java or equivalent compiler
GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent

HARDWARE:

Standalone desktops - 30 Nos.
(or)
Server supporting 30 terminals or more.

OBJECTIVES:

The student should be made to:

- Be exposed to tool kits for grid and cloud environment.
- Be familiar with developing web services/Applications in grid framework
- Learn to run virtual machines of different configuration.
- Learn to use Hadoop

LIST OF EXPERIMENTS:**GRID COMPUTING LAB**

Use Globus Toolkit or equivalent and do the following:

1. Develop a new Web Service for Calculator.
2. **Develop new OGSA-compliant Web Service.**
3. Using Apache Axis develop a Grid Service.
4. Develop applications using Java or C/C++ Grid APIs
5. **Develop secured applications using basic security mechanisms available in Globus Toolkit.**
6. Develop a Grid portal, where user can submit a job and get the result. Implement it with and without GRAM concept.

CLOUD COMPUTING LAB

Use Eucalyptus or Open Nebula or equivalent to set up the cloud and demonstrate.

1. Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time.
2. Find procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.
3. Install a C compiler in the virtual machine and execute a sample program.
4. Show the virtual machine migration based on the certain condition from one node to the other.
5. Find procedure to install storage controller and interact with it.
6. Find procedure to set up the one node Hadoop cluster.
7. Mount the one node Hadoop cluster using FUSE.
8. Write a program to use the API's of Hadoop to interact with it.
9. Write a wordcount program to demonstrate the use of Map and Reduce tasks

TOTAL: 45 PERIODS

OUTCOMES:

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

- Use the grid and cloud tool kits.
- Design and implement applications on the Grid.
- Design and Implement applications on the Cloud.

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:**SOFTWARE:**

Globus Toolkit or equivalent

Eucalyptus or Open Nebula or equivalent

HARDWARE

Standalone desktops

30 Nos

OBJECTIVES:

The student should be made to:

- Understand the challenges in parallel and multi-threaded programming.
- Learn about the various parallel programming paradigms, and solutions.

UNIT I MULTI-CORE PROCESSORS**9**

Single core to Multi-core architectures – SIMD and MIMD systems – Interconnection networks - Symmetric and Distributed Shared Memory Architectures – Cache coherence - Performance Issues – Parallel program design.

UNIT II PARALLEL PROGRAM CHALLENGES**9**

Performance – Scalability – Synchronization and data sharing – Data races – Synchronization primitives (mutexes, locks, semaphores, barriers) – deadlocks and livelocks – communication between threads (condition variables, signals, message queues and pipes).

UNIT III SHARED MEMORY PROGRAMMING WITH OpenMP**9**

OpenMP Execution Model – Memory Model – OpenMP Directives – Work-sharing Constructs - Library functions – Handling Data and Functional Parallelism – Handling Loops - Performance Considerations.

UNIT IV DISTRIBUTED MEMORY PROGRAMMING WITH MPI**9**

MPI program execution – MPI constructs – libraries – MPI send and receive – Point-to-point and Collective communication – MPI derived datatypes – Performance evaluation

UNIT V PARALLEL PROGRAM DEVELOPMENT**9**

Case studies - n-Body solvers – Tree Search – OpenMP and MPI implementations and comparison.

TOTAL: 45 PERIODS**OUTCOMES:**

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

- Program Parallel Processors.
- Develop programs using OpenMP and MPI.
- Compare and contrast programming for serial processors and programming for parallel processors.

TEXT BOOKS:

1. Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan-Kaufman/Elsevier, 2011.
2. Darryl Gove, "Multicore Application Programming for Windows, Linux, and Oracle Solaris", Pearson, 2011 (unit 2)

REFERENCES:

1. Michael J Quinn, "Parallel programming in C with MPI and OpenMP", Tata McGraw Hill, 2003.
2. Shameem Akhter and Jason Roberts, "Multi-core Programming", Intel Press, 2006.

COURSE OUTCOMES

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

- Prepare a literature survey in a specific domain as a team / individual to motivate lifelong learning.
- Identify the problem by applying acquired knowledge.
- Choose efficient tools for designing project modules.
- Design engineering solutions to complex problems utilizing a systems approach and combine all the modules for efficient testing.

GUIDELINES

1. The students are expected to get formed into a team of convenient groups of not more than 3 members on a project.
2. Every project team shall have a guide who is the member of the faculty of the institution. Identification of student group and their faculty guide has to be completed within the first two weeks from the day of beginning of 7th semester
3. The group has to identify and select the problem to be addressed as their project work. Make through literature survey and finalize a comprehensive aim and scope of their work to be done.
4. A project report has to be submitted by each student group for their project work.
5. Three reviews have to be conducted by a team of faculty (minimum of 3 and maximum of 5) along with their faculty guide as a member of faculty team (for monitoring the progress of project planning and implementation).

COURSE OUTCOMES

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

- Prepare a literature survey in a specific domain as a team / individual to motivate lifelong learning.
- Identify the problem by applying acquired knowledge
- Choose efficient tools for designing project modules
- Design engineering solutions to complex problems utilizing a systems approach and combine all the modules for efficient testing. □ Demonstrate the knowledge, skills and attitudes of a professional engineer.

TOTAL PERIODS 180

Mapping of Course Outcomes with Programme Outcomes (1/2/3 indicates strength of correlation) 3-strong, 2-Medium, 1-Weak														
COs	Programme Outcomes POs												Programme Specific Outcomes PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	2	-	-	-	-	-	-	-	3	2	3
CO2	3	3	3	3	-	-	-	-	-	-	-	1	2	2
CO3	3	2	2	1	-	-	-	-	-	-	-	1	3	3
CO4	2	2	3	1	-	-	-	-	-	-	-	1	2	3



REFERENCES:

1. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Person Education, 2007.
2. K.P. Soman, Shyam Diwakar and V. Aja, "Insight into Data Mining Theory and Practice", Eastern Economy Edition, Prentice Hall of India, 2006.
3. G. K. Gupta, "Introduction to Data Mining with Case Studies", Eastern Economy Edition, Prentice Hall of India, 2006.
4. Daniel T.Larose, "Data Mining Methods and Models", Wiley-Interscience, 2006.

CS6002

NETWORK ANALYSIS AND MANAGEMENT

L T P C

3 0 0 3

OBJECTIVES:

The student should be made to:

- Learn network devices functions and configurations hub, switch, tap and routers.
- Be familiar with network Security Devices.
- Be exposed to network services.
- Understand and analyze application performance
- Learn to analyze network traffic and protocols
- Be aware of network-troubleshooting concepts.
- Understand network security concepts.

UNIT I A SYSTEM APPROACH TO NETWORK DESIGN AND REQUIREMENT ANALYSIS

9

Introduction-**Network Service and Service based networks-** Systems and services- characterizing the services. Requirement Analysis: Concepts – Background – User Requirements- Application Requirements- Host Requirements-**Network Requirements – Requirement Analysis: Guidelines – Requirements gathering and listing-** Developing service metrics to measure performance – Characterizing behavior- developing performance threshold – **Distinguish between service performance levels. Requirement Analysis: Practice** –Template, table and maps –simplifying the requirement analysis process –case study.

UNIT II FLOW ANALYSIS: CONCEPTS, GUIDELINES AND PRACTICE

9

Background- Flows- **Data sources and sinks-** Flow models- Flow boundaries- Flow distributions- Flow specifications- **Applying the flow model-**Establishing flow boundaries-**Applying flow distributions-** Combining flow models, boundaries and distributions- Developing flow specifications-prioritizing flow- simplifying flow analysis process **–examples of applying flow specs-** case study.

UNIT III LOGICAL DESIGN: CHOICES, INTERCONNECTION MECHANISMS, NETWORK MANAGEMENT AND SECURITY

9

Background- Establishing design goals- **Developing criteria for technology evolution-** Making technology choices for design-case study- **Shared Medium-** Switching and Routing: Comparison and contrast- Switching- **Routing-Hybrid Routing/Switching Mechanisms** – Applying Interconnection Mechanism to Design – **Integrating Network management and security into the Design-** Defining Network Management- Designing with manageable resources- Network Management Architecture- Security- Security mechanism- Examples- **Network Management and security plans-** Case study.

UNIT IV NETWORK DESIGN: PHYSICAL, ADDRESSING AND ROUTING 9

Introduction- Evaluating cable plant design options – **Network equipment placement**- diagramming the physical design- **diagramming the worksheet** –case study. Introduction to Addressing and routing- establishing routing flow in the design environments- **manipulating routing flows**- developing addressing strategies- **developing a routing strategy**- case study.

UNIT V NETWORK MANAGEMENT AND SNMP PROTOCOL MODEL 9

Network and System management, Network management system platform; Current SNMP Broadband and TMN management, Network management standards. SNMPV1, SNMPV2 system architecture, **SNMPV2**, structure of management information. SNMPV2 – **MIB –** SNMPV2 protocol, SNMPV3-Architecture, **Application, MIB**, security user based security model, access control RMON.

TOTAL: 45 PERIODS

OUTCOMES:

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

- Explain the key concepts and algorithms in complex network analysis.
- Apply a range of techniques for characterizing network structure.
- Discuss methodologies for analyzing networks of different fields.
- Demonstrate knowledge of recent research in the area and exhibit technical writing and presentation skills.

TEXT BOOKS:

1. James.D.McCabe, “Practical Computer Network Analysis and Design”, 1st Edition, Morgan Kaufman, 1997.
2. Mani Subramanian, “Network Management – Principles & Practice” – 2nd Edition Prentice Hall, 2012.

REFERENCES:

1. J.Radz, “Fundamentals of Computer Network Analysis and Engineering: Basic Approaches for Solving Problems in the Networked Computing Environment”, Universe, 2005.
2. Mark Newman, “Networks: An Introduction”, Kindle Edition, 2010.
3. Laura Chappel and Gerald Combs, “Wireshark 101: Essential Skills for Network Analysis”, Kindle Edition, 2013.
4. William Stallings., “SNMP, SNMP2, SNMP3 and RMON1 and 2”, Pearson Education, 2004.
5. Daw Sudira, “Network Management”, Sonali Publications, 2004.

IT6004

SOFTWARE TESTING

**L T P C
3 0 0 3**

OBJECTIVES:

The student should be made to:

- Expose the criteria for test cases.
- Learn the design of test cases.
- Be familiar with test management and test automation techniques.
- Be exposed to test metrics and measurements.

COURSE OBJECTIVES

- To familiarize the basic data base concepts, data models and to conceptualize a database system using ER diagrams.
- To know the concepts of parallel and distributed databases.
- To familiarize the object oriented concepts in databases.
- To learn the active database concepts to enhance the data models.
- To learn the emerging technologies related to the advanced database system.

PRE - REQUISITE: Database Management System

UNIT 1 DATABASE SYSTEM CONCEPTS 9

Database System Architecture - Data Model - **Relational Model** - Entity Relationship Model - **Normalization** - Query Processing - **Query Optimization** - Transaction Processing - **Concurrency Control** - Recovery - **Database Tuning.**

UNIT II PARALLEL AND DISTRIBUTED DATABASES 9

Parallel Databases: I/O parallelism - **Inter and Intra query parallelism** - Inter and Intra Issues - operation parallelism
 - **Distributed Databases: Introduction to Distributed Database Systems** - Distributed Database
 System Architecture - Top - **Down Approach** - Distributed Database Design Fragmentation - **Allocation** - Database
 Integration - **Bottom** - up approach - **Schema Matching** - Schema Integration - **Schema Mapping**.

UNIT III OBJECT AND OBJECT RELATIONAL DATABASES 9

Concepts for object databases: Object identity - Object structure - Type constructors - OQL - Encapsulation of operations - Methods - Persistence - Type and class hierarchies - Inheritance Complex objects - Object database standards - languages and design: ODMG model - ODL - Object relational and extended - Relational systems: Object relational features in SQL / Oracle.

UNIT IV ENHANCED DATA MODELS 9

Active database concepts and triggers - Temporal databases - Spatial databases - Multimedia databases - Deductive databases - XML databases: XML data model - DTD - XML schema - XML querying - Geographic information systems.

UNIT V EMERGING TECHNOLOGIES 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 - Information retrieval - Web databases.

TOTAL PERIODS 45

COURSE OUTCOMES

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

- understand the basic database system concepts.
- design parallel and distributed databases for application development.

- apply the object oriented concepts in databases.
- design Active database concepts for enhancing the data models and for managing the Geographic information systems.
- apply the Emerging Technologies in Mobile and Web databases.

TEXT BOOKS

1. Thomas Connolly and Carolyn Begg, “Database Systems, A Practical Approach to Design, Implementation
2. M T Ozs, Patrick Valduriez, Principles of Distributed Database Systems, Prentice Hall, 1999.

REFERENCES

1. R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, Fifth Edition, Pearson Education, 2006.
2. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Fifth Edition, Tata McGraw Hill, 2006.
3. C.S.R.Prabhu, “Object oriented data base system approaches and architectures” PHI, India, 2004.
4. S. Ceri and G. Pelagati, Distributed Database System Principles and Systems, MGH, 1985.
5. Rob cornell “Data Base System And Implementation” cengage learning 2011.

WEB LINKS

1. <https://www.w3schools.in/sql/database - concepts/>
2. <http://web.cs.wpi.edu/~cs561/s12/Lectures/4 - 5/ParallelDBs.pdf>
3. <http://nptel.ac.in/courses/106106093/>

Mapping of Course Outcomes with Programme Outcomes (1/2/3 indicates strength of correlation) 3-strong, 2-Medium, 1-Weak														
COs	Programme Outcomes POs												Programme Specific Outcomes PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	2	-	-	-	-	-	-	-	1	2	3
CO2	3	2	2	3	-	-	-	-	-	-	-	3	2	2
CO3	3	2	2	3	-	-	-	-	-	-	-	1	2	3
CO4	3	2	3	2	-	-	-	-	-	-	-	1	3	3
CO5	3	2	2	2	-	-	-	-	-	-	-	1	2	3



OBJECTIVES:

The student should be made to:

- Understand the concepts of Game design and development.
- Learn the processes, mechanics and issues in Game Design.
- Be exposed to the Core architectures of Game Programming.
- Know about Game programming platforms, frame works and engines.
- Learn to develop games.

UNIT I 3D GRAPHICS FOR GAME PROGRAMMING

9

3D Transformations, Quaternions, 3D Modeling and Rendering, Ray Tracing, Shader Models, Lighting, Color, Texturing, Camera and Projections, Culling and Clipping, Character Animation, Physics-based Simulation, Scene Graphs.

UNIT II GAME ENGINE DESIGN

9

Game engine architecture, Engine support systems, Resources and File systems, Game loop and real-time simulation, Human Interface devices, Collision and rigid body dynamics, Game profiling.

UNIT III GAME PROGRAMMING

9

Application layer, Game logic, Game views, managing memory, controlling the main loop, loading and caching game data, User Interface management, Game event management.

UNIT IV GAMING PLATFORMS AND FRAMEWORKS

9

2D and 3D Game development using Flash, DirectX, Java, Python, Game engines - DX Studio, Unity.

UNIT V GAME DEVELOPMENT

9

Developing 2D and 3D interactive games using DirectX or Python – Isometric and Tile Based Games, Puzzle games, Single Player games, Multi Player games.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, students will be able to

- Discuss the concepts of Game design and development.
- Design the processes, and use mechanics for game development.
- Explain the Core architectures of Game Programming.
- Use Game programming platforms, frame works and engines.
- Create interactive Games.

TEXT BOOKS:

1. Mike Mc Shaffrly and David Graham, "Game Coding Complete", Fourth Edition, Cengage Learning, PTR, 2012.
2. Jason Gregory, "Game Engine Architecture", CRC Press / A K Peters, 2009.
3. David H. Eberly, "3D Game Engine Design, Second Edition: A Practical Approach to Real-Time Computer Graphics" 2nd Editions, Morgan Kaufmann, 2006.

REFERENCES:

1. Ernest Adams and Andrew Rollings, "Fundamentals of Game Design", 2nd Edition Prentice Hall / New Riders, 2009.
2. Eric Lengyel, "Mathematics for 3D Game Programming and Computer Graphics", 3rd Edition, Course Technology PTR, 2011.
3. Jesse Schell, The Art of Game Design: A book of lenses, 1st Edition, CRC Press, 2008.

OBJECTIVES:

The Student should be made to:

- Learn the information retrieval models.
- Be familiar with Web Search Engine.
- Be exposed to Link Analysis.
- Understand Hadoop and Map Reduce.
- Learn document text mining techniques.

UNIT I INTRODUCTION**9**

Introduction -History of IR- Components of IR - Issues –Open source Search engine Frameworks - The impact of the web on IR - The role of artificial intelligence (AI) in IR – IR Versus Web Search - Components of a Search engine- Characterizing the web.

UNIT II INFORMATION RETRIEVAL**9**

Boolean and vector-space retrieval models- Term weighting - TF-IDF weighting- cosine similarity – Preprocessing - Inverted indices - efficient processing with sparse vectors – Language Model based IR - Probabilistic IR –Latent Semantic Indexing - Relevance feedback and query expansion.

UNIT III WEB SEARCH ENGINE – INTRODUCTION AND CRAWLING**9**

Web search overview, web structure, the user, paid placement, search engine optimization/ spam. Web size measurement - search engine optimization/spam – Web Search Architectures - crawling - meta-crawlers- Focused Crawling - web indexes -- Near-duplicate detection - Index Compression - XML retrieval.

UNIT IV WEB SEARCH – LINK ANALYSIS AND SPECIALIZED SEARCH**9**

Link Analysis –hubs and authorities – Page Rank and HITS algorithms -Searching and Ranking – Relevance Scoring and ranking for Web – Similarity - Hadoop & Map Reduce - Evaluation - Personalized search - Collaborative filtering and content-based recommendation of documents and products – handling “invisible” Web - Snippet generation, Summarization, Question Answering, Cross-Lingual Retrieval.

UNIT V DOCUMENT TEXT MINING**9**

Information filtering; organization and relevance feedback – Text Mining -Text classification and clustering - Categorization algorithms: naive Bayes; decision trees; and nearest neighbor - Clustering algorithms: agglomerative clustering; k-means; expectation maximization (EM).

TOTAL: 45 PERIODS**OUTCOMES:**

Upon completion of the course, students will be able to

- Apply information retrieval models.
- Design Web Search Engine.
- Use Link Analysis.
- Use Hadoop and Map Reduce.
- Apply document text mining techniques.

UNIT II DESIGN & SOFTWARE PROCESS 9
 Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

UNIT III MODELS AND THEORIES 9
 Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.

UNIT IV MOBILE HCI 9
 Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.

UNIT V WEB INTERFACE DESIGN 9
 Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

L: 45, T: 0, TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the student should be able to:

- Design effective dialog for HCI.
- Design effective HCI for individuals and persons with disabilities.
- Assess the importance of user feedback.
- Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Web sites.
- Develop meaningful user interface.

TEXT BOOKS:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, “Human Computer Interaction”, 3rd Edition, Pearson Education, 2004 (UNIT I , II & III).
2. Brian Fling, “Mobile Design and Development”, First Edition , O’Reilly Media Inc., 2009 (UNIT –IV).
3. Bill Scott and Theresa Neil, “Designing Web Interfaces”, First Edition, O’Reilly, 2009.(UNIT-V).

CS6009	NANO COMPUTING	L T P C
		3 0 0 3

OBJECTIVES:

The student should be made to:

- Learn nano computing challenges.
- Be familiar with the imperfections.
- Be exposed to reliability evaluation strategies.
- Learn nano scale quantum computing.
- Understand Molecular Computing and Optimal Computing.

UNIT I NANOCOMPUTING-PROSPECTS AND CHALLENGES 9
 Introduction - History of Computing - Nanocomputing - Quantum Computers – Nanocomputing Technologies - Nano Information Processing - Prospects and Challenges - Physics of Nanocomputing : Digital Signals and Gates - Silicon Nanoelectronics - Carbon Nanotube Electronics - Carbon Nanotube Field-effect Transistors – Nanolithography.

UNIT II	NANOCOMPUTING WITH IMPERFECTIONS	9
Introduction - Nanocomputing in the Presence of Defects and Faults - Defect Tolerance - Towards Quadrillion Transistor Logic Systems.		
UNIT III	RELIABILITY OF NANOCOMPUTING	9
Markov Random Fields - Reliability Evaluation Strategies - NANOLAB - NANOPRISM - Reliable Manufacturing and Behavior from Law of Large Numbers.		
UNIT IV	NANOSCALE QUANTUM COMPUTING	9
Quantum Computers - Hardware Challenges to Large Quantum Computers - Fabrication, Test, and Architectural Challenges - Quantum-dot Cellular Automata (QCA) - Computing with QCA - QCA Clocking - QCA Design Rules.		
UNIT V	QCADESIGNER SOFTWARE AND QCA IMPLEMENTATION	9
Basic QCA Circuits using QCA Designer - QCA Implementation - Molecular and Optical Computing: Molecular Computing - Optimal Computing - Ultrafast Pulse Shaping and Tb/sec Data Speeds.		

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the student should be able to:

- Discuss nano computing challenges.
- Handle the imperfections.
- Apply reliability evaluation strategies.
- Use nano scale quantum computing.
- Utilize Molecular Computing and Optimal Computing.

TEXT BOOK:

1. Sahni V. and Goswami D., Nano Computing, McGraw Hill Education Asia Ltd. (2008), ISBN (13): 978007024892.

REFERNCES:

1. Sandeep K. Shukla and R. Iris Bahar., Nano, Quantum and Molecular Computing, Kluwer Academic Publishers 2004, ISBN: 1402080670.
2. Sahni V, Quantum Computing, McGraw Hill Education Asia Ltd. 2007.
3. Jean-Baptiste Waldner, Nanocomputers and Swarm Intelligence, John Wiley & Sons, Inc. 2008, ISBN (13): 978-1848210097.

IT6011

KNOWLEDGE MANAGEMENT

L T P C
3 0 0 3

OBJECTIVES:

The student should be made to:

- Learn the Evolution of Knowledge management.
- Be familiar with tools.
- Be exposed to Applications.
- Be familiar with some case studies.

REFERENCES:

1. Charles B. Fleddermann, "Engineering Ethics", Pearson Prentice Hall, New Jersey, 2004.
2. Charles E. Harris, Michael S. Pritchard and Michael J. Rabins, "Engineering Ethics – Concepts and Cases", Cengage Learning, 2009
3. John R Boatright, "Ethics and the Conduct of Business", Pearson Education, New Delhi, 2003
4. Edmund G Seebauer and Robert L Barry, "Fundamentals of Ethics for Scientists and Engineers", Oxford University Press, Oxford, 2001
5. Laura P. Hartman and Joe Desjardins, "Business Ethics: Decision Making for Personal Integrity and Social Responsibility" Mc Graw Hill education, India Pvt. Ltd., New Delhi 2013.
6. World Community Service Centre, ' Value Education', Vethathiri publications, Erode, 2011

Web sources:

1. www.onlineethics.org
2. www.nspe.org
3. www.globalethics.org
4. www.ethics.org

CS6011

NATURAL LANGUAGE PROCESSING

L T P C
3 0 0 3

OBJECTIVES:

The student should be made to:

- Learn the techniques in natural language processing.
- Be familiar with the natural language generation.
- Be exposed to machine translation.
- Understand the information retrieval techniques.

UNIT I OVERVIEW AND LANGUAGE MODELING

8

Overview: Origins and challenges of NLP-**Language and Grammar**-Processing Indian Languages-**NLP Applications**-Information Retrieval. **Language Modeling: Various Grammar**- based Language Models-**Statistical Language Model**.

UNIT II WORD LEVEL AND SYNTACTIC ANALYSIS

9

Word Level Analysis: Regular Expressions-Finite-State Automata-**Morphological Parsing**-Spelling Error Detection and **correction**-**Words and Word classes**-Part-of Speech Tagging.

Syntactic Analysis: Context-free Grammar-Constituency- Parsing-**Probabilistic Parsing**.

UNIT III SEMANTIC ANALYSIS AND DISCOURSE PROCESSING

10

Semantic Analysis: Meaning **Representation-Lexical Semantics**- **Ambiguity-Word Sense Disambiguation**. Discourse Processing: cohesion-Reference Resolution- **Discourse Coherence and Structure**.

UNIT IV NATURAL LANGUAGE GENERATION AND MACHINE TRANSLATION

9

Natural Language Generation: Architecture of NLG Systems- **Generation Tasks and Representations**- Application of NLG. **Machine Translation: Problems in Machine Translation**- Characteristics of Indian Languages- Machine Translation Approaches-**Translation involving Indian Languages**.

UNIT V INFORMATION RETRIEVAL AND LEXICAL RESOURCES

9

Information Retrieval: Design features of Information Retrieval Systems-Classical, Non-classical, Alternative Models of Information Retrieval – valuation Lexical Resources: World Net-Frame Net-Stemmers-POS Tagger- Research Corpora.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, the student should be able to:

- Analyze the natural language text.
- Generate the natural language.
- Do machine translation.
- Apply information retrieval techniques.

TEXT BOOK:

1. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.

REFERENCES:

1. Daniel Jurafsky and James H Martin, "Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", 2nd Edition, Prentice Hall, 2008.
2. James Allen, "Natural Language Understanding", 2nd edition, Benjamin /Cummings publishing company, 1995.

CS6012

SOFT COMPUTING

L T P C
3 0 0 3

OBJECTIVES:

The student should be made to:

- Learn the various soft computing frame works.
- Be familiar with design of various neural networks.
- Be exposed to fuzzy logic.
- Learn genetic programming.
- Be exposed to hybrid systems.

UNIT I INTRODUCTION

9

Artificial neural network: Introduction, characteristics- learning methods – taxonomy – Evolution of neural networks- basic models - important technologies - applications.

Fuzzy logic: Introduction - crisp sets- fuzzy sets - crisp relations and fuzzy relations: cartesian product of relation - classical relation, fuzzy relations, tolerance and equivalence relations, non-iterative fuzzy sets. Genetic algorithm- Introduction - biological background - traditional optimization and search techniques - Genetic basic concepts.

UNIT II NEURAL NETWORKS

9

McCulloch-Pitts neuron - linear separability - hebb network - supervised learning network: perceptron networks - adaptive linear neuron, multiple adaptive linear neuron, BPN, RBF, TDNN- associative memory network: auto-associative memory network, hetero-associative memory network, BAM, hopfield networks, iterative autoassociative memory network & iterative associative memory network – unsupervised learning networks: Kohonen self organizing feature maps, LVQ – CP networks, ART network.