

- the numbers up to the specified limit and another thread has to print them. Ensure proper synchronization.
12. Design, develop and run a program to implement the Banker's Algorithm. Demonstrate its working with different data values.

Instructions:

In the examination, a combination of one LEX and one YACC problem has to be asked from Part A for a total of 30 marks and one programming exercise from Part B has to be asked for a total of 20 marks.

VI SEMESTER

MANAGEMENT AND ENTREPRENEURSHIP
(Common to All Branches)

Subject Code: 10AL61	I.A. Marks : 25
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

UNIX SYSTEM PROGRAMMING

Subject Code: 10CS62	I.A. Marks : 25
Hours/Week : 04	Exam Hours: 03
Total Hours : 52	Exam Marks: 100

PART - A

UNIT – 1

6 Hours

Introduction: UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards.

UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

UNIT – 2

6 Hours

UNIX Files: File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links.

UNIT – 3**7 Hours**

UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program.

UNIT – 4**7 Hours**

UNIX Processes: The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes.

PART - B**UNIT – 5****7 Hours**

Process Control : Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups.

UNIT – 6**7 Hours**

Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.1b Timers.

Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.

UNIT – 7**6 Hours**

Interprocess Communication – 1: Overview of IPC Methods, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores.

UNIT – 8**6 Hours**

Interprocess Communication – 2: Shared Memory, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.

Text Books:

1. Terrence Chan: UNIX System Programming Using C++, Prentice Hall India, 1999.
(Chapters 1, 5, 6, 7, 8, 9, 10)
2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005.
(Chapters 7, 8, 9, 13, 14, 15)

Reference Books:

1. Marc J. Rochkind: Advanced UNIX Programming, 2nd Edition, Pearson Education, 2005.
2. Maurice J Bach: The Design of the UNIX Operating System, Pearson Education, 1987.
3. Uresh Vahalia: UNIX Internals: The New Frontiers, Pearson Education, 2001.

FILE STRUCTURES**Subject Code: 10IS63****I.A. Marks : 25****Hours/Week : 04****Exam Hours: 03****Total Hours : 52****Exam Marks: 100****PART – A****UNIT – 1****7 Hours**

Introduction: File Structures: The Heart of the file structure Design, A Short History of File Structure Design, A Conceptual Toolkit; Fundamental File Operations: Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking, Special Characters, The Unix Directory Structure, Physical devices and Logical Files, File-related Header Files, UNIX file System Commands; Secondary Storage and System Software: Disks, Magnetic Tape, Disk versus Tape; CD-ROM: Introduction, Physical Organization, Strengths and Weaknesses; Storage as Hierarchy, A journey of a Byte, Buffer Management, Input /Output in UNIX.

UNIT – 2**6 Hours****Fundamental File Structure Concepts, Managing Files of Records :**

Field and Record Organization, Using Classes to Manipulate Buffers, Using Inheritance for Record Buffer Classes, Managing Fixed Length, Fixed Field Buffers, An Object-Oriented Class for Record Files, Record Access, More about Record Structures, Encapsulating Record Operations in a Single Class, File Access and File Organization.

UNIT – 3**7 Hours**

Organization of Files for Performance, Indexing: Data Compression, Reclaiming Space in files, Internal Sorting and Binary Searching, Keysorting; What is an Index? A Simple Index for Entry-Sequenced File, Using Template Classes in C++ for Object I/O, Object-Oriented support for Indexed, Entry-Sequenced Files of Data Objects, Indexes that are too large to hold in Memory, Indexing to provide access by Multiple keys, Retrieval Using Combinations of Secondary Keys, Improving the Secondary Index structure: Inverted Lists, Selective indexes, Binding.

UNIT – 4**6 Hours**

Cosequential Processing and the Sorting of Large Files: A Model for Implementing Cosequential Processes, Application of the Model to a General Ledger Program, Extension of the Model to include Mutiway Merging, A Second Look at Sorting in Memory, Merging as a Way of Sorting Large Files on Disk.

PART - B**UNIT – 5****7 Hours**

Multi-Level Indexing and B-Trees: The invention of B-Tree, Statement of the problem, Indexing with Binary Search Trees; Multi-Level Indexing, B-Trees, Example of Creating a B-Tree, An Object-Oriented Representation of B-Trees, B-Tree Methods; Nomenclature, Formal Definition of B-Tree Properties, Worst-case Search Depth, Deletion, Merging and Redistribution, Redistribution during insertion; B* Trees, Buffering of pages; Virtual B-Trees; Variable-length Records and keys.

UNIT – 6**6 Hours**

Indexed Sequential File Access and Prefix B + Trees: Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index: Separators Instead of Keys, The Simple Prefix B+ Tree and its maintenance, Index Set Block Size, Internal Structure of Index Set Blocks: A Variable-order B- Tree, Loading a Simple Prefix B+ Trees, B-Trees, B+ Trees and Simple Prefix B+ Trees in Perspective.

UNIT – 7**7 Hours**

Hashing: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distribution, How much Extra Memory should be used?, Collision resolution by progressive overflow, Buckets, Making deletions, Other collision resolution techniques, Patterns of record access.

UNIT – 8**6 Hours**

Extendible Hashing: How Extendible Hashing Works, Implementation, Deletion, Extendible Hashing Performance, Alternative Approaches.

Text Books:

1. Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.
(Chapters 1 to 12 excluding 1.4, 1.5, 5.5, 5.6, 8.6, 8.7, 8.8)

Reference Books:

1. K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.
2. Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.
3. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.

COMPUTER NETWORKS - II**Subject Code: 10CS64****Hours/Week : 04****Total Hours : 52****I.A. Marks : 25****Exam Hours: 03****Exam Marks: 100****PART - A****UNIT - 1****6 Hours**

Packet Switching Networks - 1: Network services and internal network operation, Packet network topology, Routing in Packet networks, Shortest path routing: Bellman-Ford algorithm.

UNIT – 2**6 Hours**

Packet Switching Networks – 2: Shortest path routing (continued), Traffic management at the Packet level, Traffic management at Flow level, Traffic management at flow aggregate level.

UNIT – 3**6 Hours**

TCP/IP-1: TCP/IP architecture, The Internet Protocol, IPv6, UDP.

UNIT – 4**8 Hours**

TCP/IP-2: TCP, Internet Routing Protocols, Multicast Routing, DHCP, NAT and Mobile IP.

PART – B

UNIT - 5

7 Hours

Applications, Network Management, Network Security: Application layer overview, Domain Name System (DNS), Remote Login Protocols, E-mail, File Transfer and FTP, World Wide Web and HTTP, Network management, Overview of network security, Overview of security methods, Secret-key encryption protocols, Public-key encryption protocols, Authentication, Authentication and digital signature, Firewalls.

UNIT – 6

6 Hours

QoS, VPNs, Tunneling, Overlay Networks: Overview of QoS, Integrated Services QoS, Differentiated services QoS, Virtual Private Networks, MPLS, Overlay networks.

UNIT - 7

7 Hours

Multimedia Networking: Overview of data compression, Digital voice and compression, JPEG, MPEG, Limits of compression with loss, Compression methods without loss, Overview of IP Telephony, VoIP signaling protocols, Real-Time Media Transport Protocols, Stream control Transmission Protocol (SCTP)

UNIT – 8

6 Hours

Mobile AdHoc Networks and Wireless Sensor Networks: Overview of Wireless Ad-Hoc networks, Routing in AdHoc Networks, Routing protocols for and Security of AdHoc networks, Sensor Networks and protocol structures, Communication Energy model, Clustering protocols, Routing protocols, ZigBee technology and 802.15.4.

Text Books:

1. Communication Networks – Fundamental Concepts & key architectures, Alberto Leon Garcia & Indra Widjaja, 2nd Edition, Tata McGraw-Hill, India
(7 - excluding 7.6, 8)
2. Computer & Communication Networks, Nadir F Mir, Pearson Education, India
(9, 10 excluding 10.7, 12.1 to 12.3, 16, 17.1 to 17.6, 18.1 to 18.3, 18.5, 19, 20)

Reference Books:

1. Behrouz A. Forouzan: Data Communications and Networking, 4th Edition, Tata McGraw-Hill, 2006.
2. William Stallings: Data and Computer Communication, 8th Edition, Pearson Education, 2007.
3. Larry L Peterson and Bruce S Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.
4. Wayne Tomasi: Introduction to Data Communications and Networking, Pearson Education, 2005.

SOFTWARE TESTING

Subject Code: 10IS65
Hours/Week : 04
Total Hours : 52

I.A. Marks : 25
Exam Hours: 03
Exam Marks: 100

PART – A**UNIT 1****6 Hours**

A Perspective on Testing, Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudocode, The triangle problem, The NextDate function, The commission problem, The SATM (Simple Automatic Teller Machine) problem, The currency converter, Saturn windshield wiper.

UNIT 2**7 Hours**

Boundary Value Testing, Equivalence Class Testing, Decision Table-Based Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations.

UNIT 3**7 Hours**

Path Testing, Data Flow Testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations. Definition-Use testing, Slice-based testing, Guidelines and observations.

UNIT 4**6 Hours**

Levels of Testing, Integration Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and

system testing. A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations

PART – B

UNIT 5

7 Hours

System Testing, Interaction Testing: Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example. Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing,.

UNIT 6

7 Hours

Process Framework: Validation and verification, Degrees of freedom, Varieties of software. Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback. The quality process, Planning and monitoring, Quality goals, Dependability properties, Analysis, Testing, Improving the process, Organizational factors.

UNIT 7

6 Hours

Fault-Based Testing, Test Execution: Overview, Assumptions in fault-based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis. Test Execution: Overview, from test case specifications to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay.

UNIT 8

6 Hours

Planning and Monitoring the Process, Documenting Analysis and Test: Quality and process, Test and analysis strategies and plans, Risk planning, Monitoring the process, Improving the process, The quality team, Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.

TEXT BOOKS:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2008.
(Listed topics only from Chapters 1, 2, 5, 6, 7, 9, 10, 12, 13, 14, 15)
2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2008.
(Listed topics only from Chapters 2, 3, 4, 16, 17, 20, 24)

REFERENCE BOOKS:

1. Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008.
2. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 2nd Edition, Pearson Education, 2007.
3. Brian Marrick: The Craft of Software Testing, Pearson Education, 1995.

OPERATIONS RESEARCH

Subject Code: 10IS661
Hours/Week : 04
Total Hours : 52

I.A. Marks : 25
Exam Hours: 03
Exam Marks: 100

PART - A**UNIT – 1****6 Hours**

Introduction, Linear Programming – 1: Introduction: The origin, nature and impact of OR; Defining the problem and gathering data; Formulating a mathematical model; Deriving solutions from the model; Testing the model; Preparing to apply the model; Implementation .
Introduction to Linear Programming: Prototype example; The linear programming (LP) model.

UNIT – 2**7 Hours**

LP – 2, Simplex Method – 1: Assumptions of LP; Additional examples. The essence of the simplex method; Setting up the simplex method; Algebra of the simplex method; the simplex method in tabular form; Tie breaking in the simplex method

UNIT – 3**6 Hours**

Simplex Method – 2: Adapting to other model forms; Post optimality analysis; Computer implementation
Foundation of the simplex method.

UNIT – 4**7 Hours**

Simplex Method – 2, Duality Theory: The revised simplex method, a fundamental insight.
The essence of duality theory; Economic interpretation of duality, Primal dual relationship; Adapting to other primal forms

PART - B

UNIT – 5

7 Hours

Duality Theory and Sensitivity Analysis, Other Algorithms for LP : The role of duality in sensitive analysis; The essence of sensitivity analysis; Applying sensitivity analysis. The dual simplex method; Parametric linear programming; The upper bound technique.

UNIT – 6

7 Hours

Transportation and Assignment Problems: The transportation problem; A streamlined simplex method for the transportation problem; The assignment problem; A special algorithm for the assignment problem.

UNIT – 7

6 Hours

Game Theory, Decision Analysis: Game Theory: The formulation of two persons, zero sum games; Solving simple games- a prototype example; Games with mixed strategies; Graphical solution procedure; Solving by linear programming, Extensions.

Decision Analysis: A prototype example; Decision making without experimentation; Decision making with experimentation; Decision trees.

UNIT – 8

6 Hours

Metaheuristics: The nature of Metaheuristics, Tabu Search, Simulated Annealing, Genetic Algorithms.

Text Books:

1. Frederick S. Hillier and Gerald J. Lieberman: Introduction to Operations Research: Concepts and Cases, 8th Edition, Tata McGraw Hill, 2005.
(Chapters: 1, 2, 3.1 to 3.4, 4.1 to 4.8, 5, 6.1 to 6.7, 7.1 to 7.3, 8, 13, 14, 15.1 to 15.4)

Reference Books:

1. Wayne L. Winston: Operations Research Applications and Algorithms, 4th Edition, Cengage Learning, 2003.
2. Hamdy A Taha: Operations Research: An Introduction, 8th Edition, Pearson Education, 2007.

COMPILER DESIGN

Subject Code: 10IS662

Hours/Week : 04

Total Hours : 52

I.A. Marks : 25

Exam Hours: 03

Exam Marks: 100

PART – A

UNIT – 1

8 Hours

Introduction, Lexical analysis: Language processors; The structure of a Compiler; The evolution of programming languages; The science of building a Compiler; Applications of compiler technology; Programming language basics.

Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

UNIT – 2

6 Hours

Syntax Analysis – 1: Introduction; Context-free Grammars; Writing a Grammar. Top-down Parsing; Bottom-up Parsing.

UNIT – 3

6 Hours

Syntax Analysis – 2: Top-down Parsing; Bottom-up Parsing.

UNIT – 4

6 Hours

Syntax Analysis – 3: Introduction to LR Parsing: Simple LR; More powerful LR parsers (excluding Efficient construction and compaction of parsing tables) ; Using ambiguous grammars; Parser Generators.

PART – B

UNIT – 5

7 Hours

Syntax-Directed Translation: Syntax-directed definitions; Evaluation orders for SDDs; Applications of syntax-directed translation; Syntax-directed translation schemes.

UNIT – 6

6 Hours

Intermediate Code Generation: Variants of syntax trees; Three-address code; Translation of expressions; Control flow; Back patching; Switch-statements; Procedure calls.

UNIT – 7**6 Hours**

Run-Time Environments: Storage Organization; Stack allocation of space; Access to non-local data on the stack; Heap management; Introduction to garbage collection.

UNIT – 8**7 Hours**

Code Generation: Issues in the design of Code Generator; The Target Language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator

Text Books:

1. Alfred V Aho, Monica S.Lam, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, 2nd Edition, Pearson Education, 2007.
(Chapters 1, 3.1 to 3.4, 4 excluding 4.7.5 and 4.7.6, 5.1 to 5.4, 6.1, 6.2, 6.4, 6.6, 6.7 to 6.9, 7.1 to 7.5, 8.1 to 8.6.)

Reference Books:

1. Charles N. Fischer, Richard J. leBlanc, Jr.: Crafting a Compiler with C, Pearson Education, 1991.
2. Andrew W Apple: Modern Compiler Implementation in C, Cambridge University Press, 1997.
3. Kenneth C Loudon: Compiler Construction Principles & Practice, Cengage Learning, 1997.

DATA COMPRESSION**Subject Code: 10IS663****I.A. Marks : 25****Hours/Week : 04****Exam Hours: 03****Total Hours : 52****Exam Marks: 100****PART – A****UNIT –1****7 Hours**

Introduction, Lossless Compression -1: Compression techniques; Modeling and coding.

Mathematical preliminaries for lossless compression: Overview; Basic concepts of Information Theory; Models; Coding; Algorithmic information theory; Minimum description length principle.

Huffman coding: Overview; The Huffman coding algorithm, Minimum variance Huffman codes; Application of Huffman coding for text compression.

UNIT – 2**6 Hours**

Lossless Compression – 2: Dictionary Techniques: Overview; Introduction; Static dictionary; Adaptive dictionary; Applications: UNIX compress, GIF, PNG, V.42.

Lossless image compression: Overview; Introduction; Basics; CALIC; JPEG-LS; Multiresolution approaches; Facsimile encoding: Run-length coding, T.4 and T.6.

UNIT – 3**6 Hours**

Basics of Lossy Coding: Some mathematical concepts: Overview; Introduction; Distortion criteria; Models.

Scalar quantization: Overview; Introduction; The quantization problem; Uniform quantizer; Adaptive quantization.

UNIT – 4**7 Hours**

Vector Quantization, Differential Encoding: Vector quantization: Overview; Introduction; Advantages of vector quantization over scalar quantization; The LBG algorithm.

Differential Encoding: Overview; Introduction; The basic algorithm; Prediction in DPCM; Adaptive DPCM; Delta modulation; Speech coding; Image coding.

PART - B**UNIT – 5****7 Hours**

Some Mathematical Concepts, Transform coding: Some mathematical concepts: Linear systems; Sampling; Discrete Fourier transform; Z-transform.

Transform coding: Overview; introduction; The transform; Transforms of interest; Quantization and coding for transform coefficients; Application to image compression – JPEG; Application to audio compression – MDCT.

UNIT – 6**6 Hours**

Subband Coding, Audio Coding: Subband Coding: Overview; introduction; Filters; The basic subband coding algorithm; Bit allocation; Application to speech coding – G.722; Application to audio coding – MPEG audio; Application to image compression.

Audio Coding: Overview; Introduction; MPEG audio coding; MPEG advanced audio coding; Dolby AC3; Other standards.

UNIT – 7**6 Hours**

Wavelet-Based Compression: Overview; Introduction; Wavelets; Multiresolution and the scaling function; Implementation using Filters; Image compression; Embedded zerotree coder; Set partitioning in hierarchical trees; JPEG 2000.

UNIT – 8**7 Hours**

Video Compression: Overview; Introduction; Motion compensation; Video signal representation; H.261; Model-based coding; Asymmetric applications; MPEG-1 and MPEG-2; H.263; H.264, MPEG-4 and advanced video coding; Packet video.

Text Books:

1. Khalid Sayood: Introduction to Data Compression, 3rd Edition, Elsevier, 2006. (Chapters 1, 2 excluding 2.2.1 and 2.4.3, 3.1, 3.2, 3.2.1, 3.8.2, 5, 7.1 to 7.5, 7.6, 7.6.1, 7.6.2, 8.1 to 8.3, 8.6, 9.1 to 9.5, 10.1 to 10.4, 11, 12.6 to 12.9, 13, 14.1 to 14.4, 14.9 to 14.12, 15, 16, 18.1 to 18.13)

Reference Books:

1. D. Salomon: Data Compression: The Complete Reference, Springer, 1998.

PATTERN RECOGNITION**Subject Code: 10IS664****I.A. Marks : 25****Hours/Week : 04****Exam Hours: 03****Total Hours : 52****Exam Marks: 100****PART – A****UNIT – 1****6 Hours**

Introduction: Machine perception, an example; Pattern Recognition System; The Design Cycle; Learning and Adaptation.

UNIT – 2**7 Hours**

Bayesian Decision Theory: Introduction, Bayesian Decision Theory; Continuous Features, Minimum error rate, classification, classifiers, discriminant functions, and decision surfaces; The normal density; Discriminant functions for the normal density.

UNIT – 3**7 Hours**

Maximum-likelihood and Bayesian Parameter Estimation: Introduction; Maximum-likelihood estimation; Bayesian Estimation; Bayesian parameter estimation: Gaussian Case, general theory; Hidden Markov Models.

UNIT – 4**6 Hours**

Non-parametric Techniques: Introduction; Density Estimation; Parzen windows; k_n – Nearest- Neighbor Estimation; The Nearest- Neighbor Rule; Metrics and Nearest-Neighbor Classification.

PART – B**UNIT – 5****7 Hours**

Linear Discriminant Functions: Introduction; Linear Discriminant Functions and Decision Surfaces; Generalized Linear Discriminant Functions; The Two-Category Linearly Separable case; Minimizing the Perception Criterion Functions; Relaxation Procedures; Non-separable Behavior; Minimum Squared-Error procedures; The Ho-Kashyap procedures.

UNIT – 6**6 Hours**

Stochastic Methods: Introduction; Stochastic Search; Boltzmann Learning; Boltzmann Networks and Graphical Models; Evolutionary Methods.

UNIT – 7**6 Hours**

Non-Metric Methods: Introduction; Decision Trees; CART; Other Tree Methods; Recognition with Strings; Grammatical Methods.

UNIT – 8**7 Hours**

Unsupervised Learning and Clustering: Introduction; Mixture Densities and Identifiability; Maximum-Likelihood Estimates; Application to Normal Mixtures; Unsupervised Bayesian Learning; Data Description and Clustering; Criterion Functions for Clustering.

Text Books:

1. Richard O. Duda, Peter E. Hart, and David G. Stork: Pattern Classification, 2nd Edition, Wiley-Interscience, 2001.

Reference Books:

1. Earl Gose, Richard Johnsonbaugh, Steve Jost: Pattern Recognition and Image Analysis, PHI, Indian Reprint 2008.

COMPUTER GRAPHICS AND VISUALIZATION

Subject Code: 10IS665

Hours/Week : 04

Total Hours : 52

I.A. Marks : 25

Exam Hours: 03

Exam Marks: 100

PART - A

UNIT – 1

7 Hours

Introduction: Applications of computer graphics; A graphics system; Images: Physical and synthetic; Imaging Systems; The synthetic camera model; The programmer's interface; Graphics architectures; Programmable Pipelines; Performance Characteristics
Graphics Programming: The Sierpinski gasket; Programming Two Dimensional Applications.

UNIT – 2

6 Hours

The OpenGL: The OpenGL API; Primitives and attributes; Color; Viewing; Control functions; The Gasket program; Polygons and recursion; The three-dimensional gasket; Plotting Implicit Functions

UNIT – 3

7 Hours

Input and Interaction: Interaction; Input devices; Clients and Servers; Display Lists; Display Lists and Modeling; Programming Event Driven Input; Menus; Picking; A simple CAD program; Building Interactive Models; Animating Interactive Programs; Design of Interactive Programs; Logic Operations

UNIT – 4

6 Hours

Geometric Objects and Transformations-I: Scalars, Points, and Vectors; Three-dimensional Primitives; Coordinate Systems and Frames; Modeling a Colored Cube; Affine Transformations; Rotation, Translation and Scaling;

PART - B

UNIT – 5

5 Hours

Geometric Objects and Transformations-II: Geometric Objects and Transformations; Transformation in Homogeneous Coordinates; Concatenation of Transformations; OpenGL Transformation Matrices; Interfaces to three-dimensional applications; Quaternion's.

UNIT – 6

7 Hours

Viewing: Classical and computer viewing; Viewing with a Computer; Positioning of the camera; Simple projections; Projections in OpenGL; Hidden-

surface removal; Interactive Mesh Displays; Parallel-projection matrices; Perspective-projection matrices; Projections and Shadows.

UNIT – 7

6 Hours

Lighting and Shading: Light and Matter; Light Sources; The Phong Lighting model; Computation of vectors; Polygonal Shading; Approximation of a sphere by recursive subdivisions; Light sources in OpenGL; Specification of materials in OpenGL; Shading of the sphere model; Global Illumination.

UNIT – 8

8 Hours

Implementation: Basic Implementation Strategies; Four major tasks; Clipping; Line-segment clipping; Polygon clipping; Clipping of other primitives; Clipping in three dimensions; Rasterization; Bresenham's algorithm; Polygon Rasterization; Hidden-surface removal; Antialiasing; Display considerations.

Text Books:

1. Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson Education, 2008. (Chapters 1 to 7)

Reference Books:

1. Donald Hearn and Pauline Baker: Computer Graphics- OpenGL Version, 3rd Edition, Pearson Education, 2004.
2. F.S. Hill Jr.: Computer Graphics Using OpenGL, 3rd Edition, PHI, 2009.
3. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics, Pearson Education 1997.

PROGRAMMING LANGUAGES

Subject Code: 10IS666

Hours/Week : 04

Total Hours : 52

I.A. Marks : 25

Exam Hours: 03

Exam Marks: 100

PART - A

UNIT – 1

7 Hours

Introduction; Names, Scopes, and Bindings: The art of language design; Programming language spectrum; Why study programming languages? Compilation and interpretation; Programming environments.

Names, scope, and bindings: The notion of binding time; Object lifetime and storage management; Scope rules; Implementing scope; The meaning of names within a scope; The binding of referencing environments; Macro expansion.

UNIT – 2 **7 Hours**

Control Flow: Expression evaluation; Structured and unstructured flow; Sequencing; Selection; Iteration; Recursion; Non-determinacy

UNIT – 3 **6 Hours**

Data Types: Type systems; Type checking; Records and variants; Arrays; Strings; Sets; Pointers and recursive types; Lists; Files and Input/Output; Equality testing and assignment.

UNIT – 4 **6 Hours**

Subroutines and Control Abstraction: Review of stack layout; Calling sequences; Parameter passing; Generic subroutines and modules; Exception handling; Coroutines; Events.

PART – B

UNIT – 5 **6 Hours**

Data Abstraction and Object Orientation: Object oriented programming; Encapsulation and Inheritance; Initialization and finalization; Dynamic method binding; Multiple inheritance; Object oriented programming revisited.

UNIT – 6 **7 Hours**

Functional Languages, and Logic Languages: Functional Languages: Origins; Concepts; A review/overview of scheme; Evaluation order revisited; Higher-order functions; Functional programming in perspective. Logic Languages: Concepts; Prolog; Logic programming in perspective.

UNIT – 7 **6 Hours**

Concurrency: Background and motivation; Concurrency programming fundamentals; Implementing synchronization; Language-level mechanisms; Message passing.

UNIT – 8 **7 Hours**

Run-Time Program Management: Virtual machines; Late binding of machine code; Inspection/introspection.

Text Books:

1. Michael L. Scott: Programming Language Pragmatics, 3rd Edition, Elsevier, 2009.
(Chapters 1.1 to 1.5, 3.1 to 3.7, 6 excluding the sections on CD, 7 excluding the ML type system, 8, 9, 10 excluding the sections on CD, 11 excluding the sections on CD, 12, 15. Note: Text Boxes titled Design & Implementation are excluded)

Reference Books:

1. Ravi Sethi: Programming languages Concepts and Constructs, 2nd Edition, Pearson Education, 1996.
2. R Sebesta: Concepts of Programming Languages, 8th Edition, Pearson Education, 2008.
3. Allen Tucker, Robert Nonan: Programming Languages, Principles and Paradigms, 2nd Edition, Tata McGraw-Hill, 2007.

FILE STRUCTURES LABORATORY**Subject Code: 10ISL67****I.A. Marks : 25****Hours/Week : 03****Exam Hours: 03****Total Hours : 42****Exam Marks: 50****PART - A****Design, develop, and implement the following programs**

1. Write a C++ program to read series of names, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output.
2. Write a C++ program to read and write student objects with fixed-length records and the fields delimited by "|". Implement pack (), unpack (), modify () and search () methods.
3. Write a C++ program to read and write student objects with Variable - Length records using any suitable record structure. Implement pack (), unpack (), modify () and search () methods.
4. Write a C++ program to write student objects with Variable - Length records using any suitable record structure and to read from this file a student record using RRN.
5. Write a C++ program to implement simple index on primary key for a file of student objects. Implement add (), search (), delete () using the index.
6. Write a C++ program to implement index on secondary key, the name, for a file of student objects. Implement add (), search (), delete () using the secondary index.

7. Write a C++ program to read two lists of names and then match the names in the two lists using Cosequential Match based on a single loop. Output the names common to both the lists.
8. Write a C++ program to read k Lists of names and merge them using k-way merge algorithm with $k = 8$.
9. Write a C++ program to implement B-Tree for a given set of integers and its operations insert () and search (). Display the tree.
10. Write a C++ program to implement B+ tree for a given set of integers and its operations insert (), and search (). Display the tree.
11. Write a C++ program to store and retrieve student data from file using hashing. Use any collision resolution technique.
12. Write a C++ program to reclaim the free space resulting from the deletion of records using linked lists.

Note: In the examination *each* student picks one question from the lot of *all* 12 questions.

SOFTWARE TESTING LABORATORY

Subject Code: 10ISL68
Hours/Week : 03
Total Hours : 42

I.A. Marks : 25
Exam Hours: 03
Exam Marks: 50

1. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on decision-table approach, execute the test cases and discuss the results.
2. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, execute the test cases and discuss the results.

3. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results.
4. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
6. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of equivalence class testing, derive different test cases, execute these test cases and discuss the test results.
7. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.
8. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
9. Design, develop, code and run the program in any suitable language to implement the quicksort algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
10. Design, develop, code and run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
11. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
12. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.