V SEMESTER

SOFTWARE ENGINEERING

Subject Code: 10IS51	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Overview: Introduction: FAQ's about software engineering, Professional and ethical responsibility.
Socio-Technical systems: Emergent system properties; Systems engineering; Organizations, people and computer systems; Legacy systems.

UNIT – 2 **6 Hours**

Critical Systems, Software Processes: Critical Systems: A simple safety-critical system; System dependability; Availability and reliability.
Software Processes: Models, Process iteration, Process activities; The Rational Unified Process; Computer Aided Software Engineering.

UNIT – 3 **7 Hours**

Requirements: Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document.
Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management.

UNIT – 4 **7 Hours**

System models, Project Management: System Models: Context models; Behavioral models; Data models; Object models; Structured methods.
Project Management: Management activities; Project planning; Project scheduling; Risk management.

PART - B

UNIT – 5 **7 Hours**

Software Design: Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles.
Object-Oriented design: Objects and Object Classes; An Object-Oriented design process; Design evolution.

UNIT – 6 **6 Hours**

Development: Rapid Software Development: Agile methods; Extreme programming; Rapid application development.
Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.

UNIT – 7 **7 Hours**

Verification and Validation: Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods.
Software testing: System testing; Component testing; Test case design; Test automation.

UNIT – 8 **6 Hours**

Management: Managing People: Selecting staff; Motivating people; Managing people; The People Capability Maturity Model.
Software Cost Estimation: Productivity; Estimation techniques; Algorithmic cost modeling, Project duration and staffing.

Text Books:

1. Ian Sommerville: Software Engineering, 8th Edition, Pearson Education, 2007. (Chapters-: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 21, 22, 23, 25, 26)

Reference Books:

1. Roger.S.Pressman: Software Engineering-A Practitioners approach, 7th Edition, McGraw Hill, 2007.
2. Pankaj Jalote: An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2005.

SYSTEM SOFTWARE

Subject Code: 10CS52	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Machine Architecture: Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples.

UNIT – 2 **6 Hours**

Assemblers -1: Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation.

UNIT – 3 **6 Hours**

Assemblers -2: Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples - MASM Assembler.

UNIT – 4 **8 Hours**

Loaders and Linkers: Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Implementation Examples - MS-DOS Linker.

PART – B

UNIT – 5 **6 Hours**

Editors and Debugging Systems: Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship With Other Parts Of The System, User-Interface Criteria

UNIT – 6 **8 Hours**

Macro Processor: Basic Macro Processor Functions - Macro Definitions 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 Processor Design Options - Recursive Macro Expansion, General-Purpose Macro Processors, Macro Processing Within Language Translators, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor.

UNIT – 7 **6 Hours**

Lex and Yacc – 1: Lex and Yacc - The Simplest Lex Program, Recognizing Words With LEX, Symbol Tables, Grammars, Parser-Lexer Communication, The Parts of Speech Lexer, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Using LEX - Regular Expression, Examples of Regular Expressions, A Word Counting Program, Parsing a Command Line.

UNIT – 8 **6 Hours**

Lex and Yacc - 2

Using YACC – Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, Symbol Values and Actions, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity, Variables and Typed Tokens.

Text Books:

1. Leland.L.Beck: System Software, 3rd Edition, Addison-Wesley, 1997.
   (Chapters 1.1 to 1.3, 2 (except 2.5.2 and 2.5.3), 3 (except 3.5.2 and 3.5.3), 4 (except 4.4.3))
2. John.R.Levine, Tony Mason and Doug Brown: Lex and Yacc, O'Reilly, SPD, 1998.
   (Chapters 1, 2 (Page 2-42), 3 (Page 51-65))

Reference Books:

1. D.M.Dhamdhere: System Programming and Operating Systems, 2nd Edition, Tata McGraw - Hill, 1999.

OPERATING SYSTEMS

Subject Code: 10CS53 | **I.A. Marks : 25**
Hours/Week : 04 | **Exam Hours: 03**
Total Hours : 52 | **Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Introduction to Operating Systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special-purpose systems; Computing environments.Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

UNIT – 2 **7 Hours**

Process Management: Process concept; Process scheduling; Operations on processes; Inter-process communication. Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; Thread scheduling.

UNIT – 3 **7 Hours**

Process Synchronization : Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

UNIT – 4 **6 Hours**

Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

PART – B

UNIT – 5 **7 Hours**

Memory Management: Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

UNIT – 6 **7 Hours**

File System, Implementation of File System: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management

UNIT – 7 **6 Hours**

Secondary Storage Structures, Protection : Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of

protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability-Based systems.

UNIT – 8 **6 Hours**

Case Study: The Linux Operating System: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Inter-process communication.

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 7th edition, Wiley India, 2006.
   (Listed topics only from Chapters 1 to 12, 17, 21)

Reference Books:

1. D.M Dhamdhere: Operating systems - A concept based Approach, 2nd Edition, Tata McGraw- Hill, 2002.
2. P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
3. Harvey M Deital: Operating systems, 3rd Edition, Pearson Education, 1990.

DATABASE MANAGEMENT SYSTEMS

Subject Code: 10CS54	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **6 Hours**

Introduction: Introduction; An example; Characteristics of Database approach; Actors on the screen; Workers behind the scene; Advantages of using DBMS approach; A brief history of database applications; when not to use a DBMS.

Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment; Centralized and client-server architectures; Classification of Database Management systems.

UNIT – 2 **6 Hours**

Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design; ER Diagrams, Naming Conventions and Design Issues; Relationship types of degree higher than two.

UNIT – 3 **8 Hours**

Relational Model and Relational Algebra : Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations : JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping.

UNIT – 4 **6 Hours**

SQL – 1: SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL Queries.

PART - B

UNIT – 5 **6 Hours**

SQL – 2: Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; Additional features of SQL; Database programming issues and techniques; Embedded SQL, Dynamic SQL; Database stored procedures and SQL / PSM.

UNIT – 6 **6 Hours**

Database Design – 1: Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form

UNIT – 7 **6 Hours**

Database Design -2: Properties of Relational Decompositions; Algorithms for Relational Database Schema Design; Multivalued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form; Inclusion Dependencies; Other Dependencies and Normal Forms

UNIT – 8 **8 Hours**

Transaction Management: The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; Introduction to ARIES; The log; Other recovery-related structures; The write-ahead log protocol; Checkpointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.

Text Books:

1. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Pearson Education, 2007.
   (Chapters 1, 2, 3 except 3.8, 5, 6.1 to 6.5, 7.1, 8, 9.1, 9.2 except SQLJ, 9.4, 10)
2. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2003.
   (Chapters 16, 17.1, 17.2, 18)

Reference Books:

1. Silberschatz, Korth and Sudharshan: Data base System Concepts, 5th Edition, Mc-GrawHill, 2006.
2. C.J. Date, A. Kannan, S. Swamynatham: A Introduction to Database Systems, 8th Edition, Pearson education, 2006.

COMPUTER NETWORKS - I

Subject Code: 10CS55 **I.A. Marks : 25**
Hours/Week : 04 **Exam Hours: 03**
Total Hours : 52 **Exam Marks: 100**

PART – A

UNIT – 1 **8 Hours**

Communication Networks and Services, Applications and Layered Architectures : Evolution of Network Architecture and Services; Future network architectures and their services; Key factors in communication network evolution

Examples of Protocols, Services, and Layering; The OSI Reference Model; Overview of TCP/IP Architecture; Application Layer Protocols and TCP/IP Utilities.

UNIT – 2 **6 Hours**

Digital Transmission – 1: Digital Representation of Information: Block-Oriented Information, Stream Information; Why Digital Communications? Comparison of Analog and Digital Transmission , Basic properties of Digital Transmission Systems; Digital Representation of Analog Signals: Bandwidth of Analog Signals, Sampling of an Analog Signal, Digital Transmission of Analog Signals; Characterization of Communication Channels: Frequency Domain Characterization, Time Domain Characterization; Fundamental Limits in Digital Transmission: The Nyquist Signaling Rate, The Shannon Channel Capacity; Line Coding.

UNIT – 3 **6 Hours**

Digital Transmission – 2: Modems and Digital Modulation: Binary Phase Modulation, QAM and Signal Constellations, Telephone Modem Standards; Properties of Media and Digital Transmission Systems: Twisted Pair, Coaxial Cable, Optical Fiber, Radio Transmission, Infrared Light; Error Detection and Correction: Error Detection, Two Dimensional Parity Checks, Internet Checksum, Polynomial Codes, Standardized Polynomial Codes, Error Detecting Capability of a Polynomial Code.

UNIT – 4 **6 Hours**

Circuit Switching Networks: Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing, Wavelength-Division Multiplexing; SONET: SONET Multiplexing, SONET Frame Structure; Transport Networks: SONET Networks, Optical Transport networks; Circuit Switches: Space Division Switches, Time Division Switches; The Telephone Network: Transmission Facilities, End to End Digital Services; Cellular telephone networks.

PART - B

UNIT – 5 **6 Hours**

Peer-to-Peer Protocols and Data Link Layer - I : Peer-to-Peer Protocols and Service Models: Service models, Examples of services, End to end versus hop by hop; ARQ Protocols and Reliable Data Transfer Service: Stop-and-Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ; Other Peer-to-Peer Protocols: Sliding-window flow control, Timing recovery for synchronous services, TCP reliable stream service and flow control.

UNIT – 6 **6 Hours**

Peer-to-Peer Protocols and Data Link Layer – II: Data Link Controls: Framing; Point to Point Protocol; HDLC Data link Control: Data link services, HDLC configuration and transfer modes, HDLC frame format, Typical frame exchanges; Link Sharing using Packet Multiplexers: Statistical Multiplexing, Speech Interpolation and the Multiplexing of Packetized Speech.

UNIT – 7 **7 Hours**

Medium Access Control Protocols and Local Area Networks – I: The Medium Access Control Protocols: Multiple Access Communications; Random Access : ALOHA, Slotted ALOHA, CSMA, CSMA-CD; Scheduling Approaches to Medium Access Control: Reservation Systems, Polling, Token-Passing Rings, Comparison of scheduling approaches in MAC, Comparison of random access and scheduling MAC; Channelization: FDMA, TDMA, CDMA.

UNIT – 8 **7 Hours**

Medium Access Control Protocols and Local Area Networks – II : LAN Protocols: LAN Structure , The Medium Access Control Sublayer, The Logical Link Control Sublayer; Ethernet and IEEE 802.3 LAN Standard: Ethernet Protocol, Frame structure, Physical Layers, Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet; Token Ring and IEEE 802.5 LAN Standard: Token-Ring Protocol, Frame structure; FDDI; Wireless LANs and IEEE 802.11 Standard: Ad hoc and Infrastructure Networks, Frame structure and addressing, Medium Access Control; LAN Bridges and Ethernet Switches: Transparent Bridges, Source Routing Bridges, Mixed-Media Bridges, Virtual LANs.

Text Books:

1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks -Fundamental Concepts and Key architectures, 2nd Edition Tata McGraw-Hill, 2004.
   (Chapters 1, 2.1 to 2.3, 2.5, 3.1 to 3.9 except 3.3.4, 3.9.7 and 3.9.8 , 4.1 to 4.5, 4.8, 5, 6.1 to 6.4, 6 except 6.10.4)

References:

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. David: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.
4. Nader F. Mir: Computer and Communication Networks, Pearson Education, 2007.

FORMAL LANGUAGES AND AUTOMATA THEORY

Subject Code: 10CS56	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **7 Hours**

Introduction to Finite Automata: Introduction to Finite Automata; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata

UNIT – 2 **7 Hours**

Finite Automata, Regular Expressions: An application of finite automata; Finite automata with Epsilon-transitions; Regular expressions; Finite Automata and Regular Expressions; Applications of Regular Expressions

UNIT – 3 **6 Hours**

Regular Languages, Properties of Regular Languages: Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata

UNIT – 4 **6 Hours**

Context-Free Grammars And Languages : Context –free grammars; Parse trees; Applications; Ambiguity in grammars and Languages .

PART – B

UNIT – 5 **7 Hours**

Pushdown Automata: Definition of the Pushdown automata; the languages of a PDA; Equivalence of PDA's and CFG's; Deterministic Pushdown Automata

UNIT – 6 **6 Hours**

Properties of Context-Free Languages: Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFLs

UNIT – 7 **7 Hours**

Introduction To Turing Machine: Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines; Extensions to the basic Turning Machines; Turing Machine and Computers.

UNIT – 8 **6 Hours**

Undecidability: A Language that is not recursively enumerable; An Undecidable problem that is RE; Post's Correspondence problem; Other undecidable problems.

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Education, 2007.
   (Chapters: 1.1, 1.5, 2.2 to 2.5, 3.1 to 3.3, 4, 5, 6, 7, 8.1 to 8.4, 8.6, 9.1, 9.2, 9.4.1, 9.5)

Reference Books:

1. K.L.P. Mishra: Theory of Computer Science, Automata, Languages, and Computation, 3rd Edition, PHI, 2007.
2. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation, Principles and Practice, Morgan Kaufmann, 1998.
3. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
4. Thomas A. Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition, Pearson Education, 2006.

DATABASE APPLICATIONS LABORATORY

Subject Code: 10CSL57	**I.A. Marks : 25**
Hours/Week : 03	**Exam Hours: 03**

Total Hours : 42 **Exam Marks: 50**

1. Consider the following relations:
   Student (*snum:* integer, *sname:* string, *major:* string, *level:* string, *age:* integer)
   Class (*name:* string, *meets at:* string, *room:* string, *d:* integer)
   Enrolled (*snum:* integer, *cname:* string)
   Faculty (fi*d*: integer, *fname:* string, *deptid:* integer)
   The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc)
   Write the following queries in SQL. No duplicates should be printed in any of the answers.
   i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
   ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
   iii. Find the names of all students who are enrolled in two classes that meet at the same time.
   iv. Find the names of faculty members who teach in every room in which some class is taught.
   v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

2. The following relations keep track of airline flight information:
   Flights (*no:* integer, *from:* string, *to:* string, *distance:* integer, *Departs:* time, *arrives:* time, *price:* real)
   Aircraft (*aid:* integer, *aname:* string, *cruisingrange:* integer)
   Certified (*eid:* integer, *aid:* integer)
   Employees (*eid:* integer, *ename:* string, *salary:* integer)
   Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.
   Write each of the following queries in SQL.

   i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
   ii. For each pilot who is certified for more than three aircrafts, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.
   iii. Find the names of pilots whose *salary* is less than the price of the cheapest route from Bengaluru to Frankfurt.
   iv. For all aircraft with *cruisingrange* over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.
   v. Find the names of pilots certified for some Boeing aircraft.
   vi. Find the *aid*s of all aircraft that can be used on routes from Bengaluru to New Delhi.

3. Consider the following database of student enrollment in courses & books adopted for each course.
   STUDENT (regno: string, name: string, major: string, bdate:date)
   COURSE (course #:int, cname:string, dept:string)
   ENROLL (regno:string, course#:int, sem:int, marks:int)
   BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)
   TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)
   i. Create the above tables by properly specifying the primary keys and the foreign keys.
   ii. Enter at least five tuples for each relation.
   iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
   iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
   v. List any department that has *all* its adopted books published by a specific publisher.
   vi. Generate suitable reports.
   vii. Create suitable front end for querying and displaying the results.

4. The following tables are maintained by a book dealer.
   AUTHOR (author-id:int, name:string, city:string, country:string)
   PUBLISHER (publisher-id:int, name:string, city:string, country:string)
   CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)
   CATEGORY (category-id:int, description:string)

ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
iv. Find the author of the book which has maximum sales.
v. Demonstrate how you increase the price of books published by a specific publisher by 10%.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

5. Consider the following database for a banking enterprise

BRANCH(branch-name:string, branch-city:string, assets:real)
ACCOUNT(accno:int, branch-name:string, balance:real)
DEPOSITOR(customer-name:string, accno:int)
CUSTOMER(customer-name:string, customer-street:string, customer-city:string)
LOAN(loan-number:int, branch-name:string, amount:real)
BORROWER(customer-name:string, loan-number:int)

i. Create the above tables by properly specifying the primary keys and the foreign keys
ii. Enter at least five tuples for each relation
iii. Find all the customers who have at least two accounts at the *Main* branch.
iv. Find all the customers who have an account at *all* the branches located in a specific city.
v. Demonstrate how you delete all account tuples at every branch located in a specific city.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

Instructions:

1. The exercises are to be solved in an RDBMS environment like Oracle or DB2.
2. Suitable tuples have to be entered so that queries are executed correctly.
3. Front end may be created using either VB or VAJ or any other similar tool.
4. The student need not create the front end in the examination. The results of the queries may be displayed directly.
5. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.
6. Questions must be asked based on lots.

SYSTEM SOFTWARE & OPERATING SYSTEMS LABORATORY

Subject Code: 10CSL58	**I.A. Marks : 25**
Hours/Week : 03	**Exam Hours: 03**
Total Hours : 42	**Exam Marks: 50**

PART - A

LEX and YACC Programs:

Design, develop, and execute the following programs using LEX:

1. a) Program to count the number of characters, words, spaces and lines in a given input file.
   b) Program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.

2. a) Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.
   b) Program to recognize whether a given sentence is simple or compound.

3. Program to recognize and count the number of identifiers in a given input file.

   Design, develop, and execute the following programs using YACC:

4. a) Program to recognize a valid arithmetic expression that uses operators +, -, * and /.

b) Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.

5. a) Program to evaluate an arithmetic expression involving operators +, -, * and /.
   b) Program to recognize strings 'aaab', 'abbb', 'ab' and 'a' using the grammar (a^nb^n, n>= 0).

6. Program to recognize the grammar (a^nb, n>= 10).

PART B

UNIX Programming:

Design, develop, and execute the following programs:

7. a) Non-recursive shell script that accepts any number of arguments and prints them in the Reverse order, (For example, if the script is named rargs, then executing rargs A B C should produce C B A on the standard output).
   b) C program that creates a child process to read commands from the standard input and execute them (a minimal implementation of a shell – like program). You can assume that no arguments will be passed to the commands to be executed.

8. a) Shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, outputs the common permissions, otherwise outputs each file name followed by its permissions.
   b) C program to create a file with 16 bytes of arbitrary data from the beginning and another 16 bytes of arbitrary data from an offset of 48. Display the file contents to demonstrate how the hole in file is handled.

9. a) Shell script that accepts file names specified as arguments and creates a shell script that contains this file as well as the code to recreate these files. Thus if the script generated by your script is executed, it would recreate the original files(This is same as the "bundle" script described by Brain W. Kernighan and Rob Pike in " The Unix Programming Environment", Prentice – Hall India).
   b) C program to do the following: Using fork( ) create a child process. The child process prints its own process-id and id of its parent and then exits. The parent process waits for its child to finish (by executing the wait( )) and prints its own process-id and the id of its child process and then exits.

Operating Systems:

10. Design, develop and execute a program in C / C++ to simulate the working of Shortest Remaining Time and Round-Robin Scheduling Algorithms. Experiment with different quantum sizes for the Round-Robin algorithm. In all cases, determine the average turn-around time. The input can be read from key board or from a file.
11. Using OpenMP, Design, develop and run a multi-threaded program to generate and print Fibonacci Series. One thread has to generate 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 SYSTEMS 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.lb 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, Pearson Education, 2001.

COMPILER DESIGN

Subject Code: 10CS63 / 10IS662	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Introduction, Lexical analysis: Compilers; Analysis of Source Program; The Phases of a Compiler; Cousins of the Compiler; The grouping of phases; Compiler- Construction tools.
Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

UNIT – 2 **7 Hours**

Syntax Analysis – 1: The Role of the Parser; Context-free Grammars; Writing a Grammar; Top-down Parsing; Bottom-up Parsing.

UNIT – 3 **7 Hours**

Syntax Analysis – 2
Operator-Precedence Parsing; LR Parsers; Using ambiguous grammars; Parser Generators.

UNIT – 4 **6 Hours**

Syntax-Directed Translation: Syntax-Directed definitions; Constructions of Syntax Trees; Bottom-up evaluation of S-attributed definitions; L-attributed definitions; Top-down translation.

PART – B

UNIT – 5 **6 Hours**

Run-Time Environments: Source Language Issues; Storage Organization; Storage-allocation strategies, Storage-allocation in C; Parameter passing

UNIT – 6 **6 Hours**

Intermediate Code Generation: Intermediate Languages; Declarations; Assignment statements; Boolean Expressions; Case statements; Back patching; Procedure calls.

UNIT – 7 **7 Hours**

Code Generation: Issues in the design of Code Generator; The Target Machine; Run-time Storage Management; Basic blocks and Flow graphs; Next-use information; A Simple Code Generator; Register allocation and assignment; The dag representation of basic blocks; Generating code from dags.

UNIT – 8 **7 Hours**

Code Optimization, Compiler Development: Code Optimization: Introduction; The principal sources of optimization; Peephole optimization; Optimization of basic blocks; Loops in flow graphs.
Compiler Development: Planning a compiler; Approaches to compiler development; the compiler development environment; Testing and maintenance.

Text Books:

1. Alfred V Aho, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, Pearson Education, 2007.
(Chapters 1, 3.1 to 3.4, 4, 5.1 to 5.5, 7, 8, 9.1 to 9.9, 10.1 to 10.5, 11)

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 Louden: Compiler Construction Principles & Practice, Thomson Education, 1997.

COMPUTER NETWORKS - II

Subject Code: 10CS64 **I.A. Marks : 25**
Hours/Week : 04 **Exam Hours: 03**
Total Hours : 52 **Exam Marks: 100**

PART - A

UNIT – 1 **6 Hours**

Packet-Switching Networks – 1: Network services and internal network operations; Packet network topology; Datagrams and virtual circuits; Routing in packet networks; Shortest-path routing; ATM networks

UNIT – 2 **6 Hours**

Packet-Switching Networks – 2, TCP / IP – 1: Traffic management at the packet level; Traffic management at the flow level; Traffic management at the flow-aggregate level
The TCP / IP architecture; The Internet protocol

UNIT – 3 **7 Hours**

TCP / IP – 2: IPv6; User datagram protocol; Transmission control protocol; Internet routing protocols; Multicast routing; DHCP, NAT, and Mobile IP

UNIT – 4 **7 Hours**

ATM Networks: Why ATM? BISDN reference model; ATM layer; ATM adaptation layer; ATM signaling; PNNI routing; Classical IP over ATM

PART – B

UNIT – 5 **6 Hours**

Network Management, Security: Network management overview; SNMP; Structure of Management information; MIB; Remote network monitoring
Security and cryptographic algorithms; Security protocols; Cryptographic algorithms

UNIT – 6 **7 Hours**

QoS, Resource Allocation, VPNs, Tunneling, Overlay Networks: Overview of QOS; Integrated services QoS; Differentiated services QoS; Resource allocation.
Virtual Private Networks; Multiprotocol Label switching; Overlay networks

UNIT – 7 **7 Hours**

Compression of Digital Voice and Video, VoIP, Multimedia Networking: Overview of data compression; Digital voice and compression; Still images and JPEG compression; Moving images and MPEG compression; Limits of compression with loss; Compression methods without loss; Case Study: FAX compression for transmission.
Overview of IP telephony; VoIP signaling protocols; Real-Time media transport protocols; Distributed multimedia networking; SCTP

UNIT – 8 **6 Hours**

Mobile Ad-Hoc Networks, Wireless sensor Networks: Overview of wireless adhoc networks; Routing in adhoc networks; Routing protocols for adhoc networks; security of adhoc networks. Sensor networks and protocol structures; Communication energy model; Clustering protocols; Routing protocols; Zigbee technology and IEEE 802.15.4

Text Books:

1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks – Fundamental Concepts and Key architectures, 2nd Edition, Tata McGraw-Hill, 2004.
   (Chapters 7, 8, 9, 11, Appendix B)
2. Nader F. Mir: Computer and Communication Networks, Pearson Education, 2007.
   (Chapters 12, 16, 17, 18, 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. David: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.
4. Wayne Tomasi: Introduction to Data Communications and Networking, Pearson Education, 2005.

COMPUTER GRAPHICS AND VISUALIZATION

Subject Code: 10CS65 / 10IS665	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**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, 2nd Edition, Pearson education, 2001.
3. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics, Addison-Wesley 1997.

OPERATIONS RESEARCH

Subject Code: 10CS661/10IS661	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**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, 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, Thomson Course Technology, 2003.
2. Hamdy A Taha: Operations Research: An Introduction, 8th Edition, Prentice Hall India, 2007.

SIGNALS AND SYSTEMS

Subject Code: 10CS662	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **7 Hours**
Introduction: Definitions of a signal and a system; Classification of signals; Basic operations on signals; Elementary signals.

UNIT – 2 **7 Hours**
Systems, Time-domain representations – 1: Systems viewed as interconnections of operations; Properties of systems; Convolution; Impulse response representation; Properties of impulse response representation.

UNIT – 3 **6 Hours**
Time domain representation – 2: Differential and difference equation representations; Block diagram representations.

UNIT – 4 **6 Hours**
Fourier Representation – 1: Fourier representation: Introduction; Fourier representations for four signal classes; Orthogonality of complex sinusoidal signals.

PART – B

UNIT – 5 **6 Hours**
Fourier Representation -2: DTFS representations; Continuous-tine Fourier-series representations; DTFT and FT representations; Properties of Fourier representations.

UNIT – 6 **7 Hours**
Application of Fourier representations – 1: Frequency response of LTI systems; Solution of differential and difference equations using system function.

UNIT – 7 **7 Hours**
Applications of Fourier Representations – 2, Z-Transforms – 1: Fourier transform representations for periodic signals; Sampling of continuous time signals and signal reconstruction.
Introduction to Z-transform; Properties of ROC; Properties of Z-transforms; Inversion of Z-transforms

UNIT –Z – 8 **6 Hours**

Transforms – 2: Transforms analysis of LTI systems; Transfer function; Stability and causality; Unilateral Z-transforms and its application to solve difference equations

Text Books:

1. Simon Haykin and Barry Van Veen: Signals and Systems, John Wiley and Sons, 2001, Reprint 2002. (Chapters: 1.1 to 1.8, 2.2 to 2.5, 3.1 to 3.6, 4.2 to 4.3, 4.7, 7.1 to 7.6, 7.8)

Reference Books:

1. Alan V. Oppenheim, Alan S. Willsky and S. Hamid Nawab: Signals and Systems, Pearson Education Asia, 2nd Edition, 1997, Indian reprint 2002.
2. Ganesh Rao D and Satish Tunga: Signals and Systems - A Simplified Approach, Sanguine Technical Publishers, 2003-04.

DATA COMPRESSION

Subject Code: 10CS663/10IS663	**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, Minimumvariance 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; Multiresoution 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: 10CS664/10IS664	**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, Pearson Education, 2007.

STOCHASTIC MODELS AND APPLICATIONS

Subject Code: 10CS665	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Introduction – 1: Axioms of probability; Conditional probability and independence; Random variables; Expected value and variance; Moment-Generating Functions and Laplace Transforms; conditional expectation; Exponential random variables.

UNIT – 2 **6 Hours**

Introduction – 2: Limit theorems; Examples: A random graph; The Quicksort and Find algorithms; A self-organizing list model; Random permutations.

UNIT – 3 **7 Hours**

Probability Bounds, Approximations, and Computations: Tail probability inequalities; The second moment and conditional expectation inequality; probability bounds via the Importance sampling identity; Poisson random variables and the Poisson paradigm; Compound Poisson random variables.

UNIT – 4 **7 Hours**

Markov Chains: Introduction; Chapman-Kologorov Equations; Classification of states; Limiting and stationary probabilities; some applications; Time-Reversible Markov Chains; Markov Chain Monte Carlo methods.

PART – B

UNIT – 5 **6 Hours**

The Probabilistic Method: Introduction; Using probability to prove existence; Obtaining bounds from expectations; The maximum weighted independent set problem: A bound and a ranom algorithm; The set covering problem; Antichains; The Lovasz Local lemma; A random algorithm for finding the minimal cut in a graph.

UNIT – 6 **6 Hours**

Martingales: Martingales: Definitions and examples; The martingale stopping theorem; The Hoeffding-Azuma inequality; Sub-martingales.

UNIT – 7 **7 Hours**

Poisson Processes, Queuing Theory – 1: The non-stationary Poisson process; The stationary Poisson process; Some Poisson process computations; Classifying the events of a non-stationary Poisson process; Conditional distribution of the arrival times

Queuing Theory: Introduction; Preliminaries; Exponential models

UNIT – 8 **7 Hours**

Queuing Theory – 2: Birth-and-Death exponential queuing systems; The backwards approach in exponential queues; A closed queuing network; An open queuing network; The M/G/1 queue; Priority queues.

Text Books:

1. Sheldon M. Ross: Probability Models for Computer Science, Elsevier, 2002.

Reference Books:

1. B. R. Bhat: Stochastic Models Analysis and Applications, New Age International, 2000.
2. Scott L. Miller, Donald G. Childers: Probability and Random Processes with Applications to Signal Processing and Communications, Elsevier, 2004.

PROGRAMMING LANGUAGES

Subject Code: 10CS666/10IS666	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **7 Hours**

Introduction; Names, Scope, and Bindings – 1: Language design; Programming language spectrum; why study programming languages? Compilation and interpretation; Programming environments.
Names, scope, and bindings: Concept of binding time; Object lifetime and storage management; Scope rules and implementing scope.

UNIT – 2 **7 Hours**

Names, Scope, and Bindings – 1; Control Flow – 1 : The binding of reference environments; Binding within a scope; Separate compilation.Control Flow – 1: Expression evaluation.

UNIT – 3 **6 Hours**

Control Flow – 2: Structured and unstructured flow; Sequencing; Selection; Iteration; Recursion; Non-determinacy

UNIT – 4 **6 Hours**

Data Types – 1: Type systems; Type checking; Records and variants; Arrays

PART – B

UNIT – 5 **7 Hours**

Data Types – 2: Strings; Sets; Pointers and recursive types; Lists; Files and Input/Output; Equality testing and assignment

UNIT – 6 **6 Hours**

Subroutines and Control Abstraction – 1: Review of stack layout; Calling sequences; Parameter passing; Generic subroutines and modules; Exception handling.

UNIT – 7 **6 Hours**

Control Abstraction – 2; Data Abstraction, Object Orientation: Control abstraction – 2: Coroutines
Data Abstraction, Object Orientation: Object oriented programming; Encapsulation and Inheritance; Dynamic method binding; Multiple inheritance; Object oriented programming revisited

UNIT – 8 **7 Hours**

Functional Languages, Logic Languages, Scripting Languages: Functional Labguages: Origins; Concepts; An overview of scheme; Evaluation order revisited; Higher-order functions; Functional programming in perspective.
Logic Languages: Concepts; Prolog; Logic programming in perspective.
Scripting Languages: Common characteristics

Text Books:

1. Michael L. Scott: Programming Language Pragmatics, 2nd Edition, Elsevier, 2006.

(Chapters 1.1 to 1.5, 3 excluding the sections on CD, 6 excluding the sections on CD, 7 including the sections on CD, 8 excluding the sections on CD, 9 including the sections on CD, 10 excluding the sections on CD, 11 excluding the sections on CD, 13.1. 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, Tata McGraw-Hill, 2002.

COMPUTER GRAPHICS AND VISUALIZATION LABORATORY

Subject Code: 10CSL67	**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 in C / C++

1. Program to recursively subdivide a tetrahedron to from 3D Sierpinski gasket. The number of recursive steps is to be specified by the user.
2. Program to implement Liang-Barsky line clipping algorithm.
3. Program to draw a color cube and spin it using OpenGL transformation matrices.
4. Program to create a house like figure and rotate it about a given fixed point using OpenGL functions.
5. Program to implement the Cohen-Sutherland line-clipping algorithm. Make provision to specify the input line, window for clipping and view port for displaying the clipped image.
6. Program to create a cylinder and a parallelepiped by extruding a circle and quadrilateral respectively. Allow the user to specify the circle and the quadrilateral.
7. Program, using OpenGL functions, to draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the properties of the surfaces of the solid object used in the scene.
8. Program to draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing. Use OpenGL functions.
9. Program to fill any given polygon using scan-line area filling algorithm. (Use appropriate data structures.)
10. Program to display a set of values {fij} as a rectangular mesh.

PART - B

Develop a suitable Graphics package to implement the skills learnt in the theory and the exercises indicated in Part A. Use the OpenGL.

Note:

1. Any question from Part A may be asked in the examination.
2. A report of about 10 – 12 pages on the package developed in Part B, duly certified by the department must be submitted during examination.

Instructions:

In the examination, one exercise from Part A is to be asked for a total of 30 marks. The package developed under Part B has to be evaluated for a total of 20 marks.

UNIX SYSTEMS PROGRAMMING AND COMPILER DESIGN LABORATORY

Subject Code: 10CSL68	**I.A. Marks : 25**
Hours/Week : 03	**Exam Hours: 03**
Total Hours : 42	**Exam Marks: 50**

List of Experiments for USP: Design, develop, and execute the following programs

1. Write a C/C++ POSIX compliant program to check the following limits:
   (i) No. of clock ticks (ii) Max. no. of child processes (iii) Max. path length
   (iv) Max. no. of characters in a file name (v) Max. no. of open files/ process

2. Write a C/C++ POSIX compliant program that prints the POSIX defined configuration options supported on any given system using feature test macros.

3. Consider the last 100 bytes as a region. Write a C/C++ program to check whether the region is locked or not. If the region is locked, print pid of the process which has locked. If the region is not locked, lock the region with an exclusive lock, read the last 50 bytes and unlock the region.

4. Write a C/C++ program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

5. a) Write a C/C++ program that outputs the contents of its Environment list
   b) Write a C / C++ program to emulate the unix **ln** command

6. Write a C/C++ program to illustrate the race condition.

7. Write a C/C++ program that creates a zombie and then calls system to execute the **ps** command to verify that the process is zombie.

8. Write a C/C++ program to avoid zombie process by forking twice.

9. Write a C/C++ program to implement the **system** function.

10. Write a C/C++ program to set up a real-time clock interval timer using the **alarm** API.

List of Experiments for Compiler Design: Design, develop, and execute the following programs.

11. Write a C program to implement the syntax-directed definition of “if E then S1” and “if E then S1 else S2”. (Refer Fig. 8.23 in the text book prescribed for 06CS62 Compiler Design, Alfred V Aho, Ravi Sethi, and Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, Addison-Wesley, 2007).

12. Write a yacc program that accepts a regular expression as input and produce its parse tree as output.

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