

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.

COMPILER DESIGN

Subject Code: 10CS63

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.

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 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.

COMPUTER GRAPHICS AND VISUALIZATION

Subject Code: 10CS65
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.

OPERATIONS RESEARCH

Subject Code: 10CS661
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.

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-time 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, 2nd Edition, Wiley India, 2007.
(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:

PROGRAMMING LANGUAGES

Subject Code: 10CS666
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 Sebasta: 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.

COMPUTER GRAPHICS AND VISUALIZATION LABORATORY

Subject Code: 10CSL67
Hours/Week : 03
Total Hours : 42

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

PART - A

Design, develop, and implement the following programs in C / C++

1. Program to recursively subdivide a tetrahedron to form 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 $\{f_{ij}\}$ 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
Hours/Week : 03
Total Hours : 42

I.A. Marks : 25
Exam Hours: 03
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, 2nd Edition, Pearson Education, 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.