NAGARJUNA

COLLEGE OF ENGINEERING & TECHNOLOGY

An Autonomous College under VTU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION

Excellence in creating globally competent professionals and moulding them as leaders in Computer Science & Engineering education and research.

MISSION

M1: Maintaining excellence in Computer Science & Engineering education through academic professionalism, teaching, curricula which reflect the changing needs of the society.

M2: Establishing centre of excellence by creating knowledge through research and industrial exposure in the area of Computer Science & Engineering.

M3: Developing communication skill, leadership qualities, team work & skills for continuing education among the students.

M4: Inculcating ethics, human values and skills for solving societal problems and environmental protection.

M5: Validate engineering knowledge through innovative research projects to enhance their employability and entrepreneurship skills.

III to VIII Semesters

Outcome Based Education (OBE) / Choice Based Credit System (CBCS) Curricula

With effect from Academic Year

2016 - 17

Program Educational Objectives (PEOs)

The graduates of Computer Science and Engineering are expected to fulfill the following PEOs after a few years of their graduation.

PEO1	Graduates in Computer Science and Engineering will apply the technical knowledge of analysis and design of software used for sustainable societal growth.
PEO2	Graduates of Computer Science and Engineering will demonstrate logical thinking and programming skills.
PEO3	Graduates in Computer Science and Engineering will demonstrate good communication skills, dynamic leadership qualities with concern for environmental protection.
PEO4	Computer Science and Engineering graduates will be capable of pursuing higher studies, take up research and development work blended with ethics and human values.
PEO5	Computer Science and Engineering graduates will have the ability to become entrepreneurs there by switching over from responsive engineer to creative engineer.

Program Outcomes (POs)

PO1	**Engineering Knowledge:** Apply knowledge of mathematics and science, with fundamentals of Computer Science and Engineering to be able to solve complex engineering problems related to CSE.
PO2	**Problem Analysis:** Identify, Formulate, review research literature and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO3	**Design/ Development of solutions:** Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural societal and environmental considerations.
PO4	**Conduct Investigations of Complex problems:** Use research–based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to computer science related complex engineering activities with an understanding of the limitations.

PO6	**The Engineer and Society:** Apply Reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CSE professional engineering practice.
PO7	**Environment and Sustainability:** Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply Ethical Principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary Settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large such as able to comprehend and with write effective reports and design documentation, make effective presentations and give and receive clear instructions.
PO11	**Project Management and Finance:** Demonstrate knowledge and to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
PO12	**Life-Long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning the broadest context of technological change.

Program Specific Outcome (PSOs)

PSO1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.
PSO2	**Problem - Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO3	**Mathematical concepts:** Ability to apply mathematical concepts to solve real world problems using appropriate data structure and suitable algorithms.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY
Department of Computer Science & Engineering
Outcome Based Education (OBE)/Choice Based Credit System (CBCS)
Third Semester B.E.- Scheme

Sl. No	Course Code	Course	Teaching Dept.	L-T-P-S (Hrs/week)	Total Credits	Marks
1	15CSM31	Engineering Mathematics-III (IC)	Mathematics	3-0-2-0	4	100
2	15CST32	Fundamentals of Computation Engineering	CSE	3-0-0-0	3	100
3	15CSI33	Data Structures with C (IC)	CSE	3-0-2-4	5	100
4	15CST34	Analog and Digital Electronics	CSE	3-0-0-0	3	100
5	15CST35	Computer Organization	CSE	3-0-0-0	3	100
6	15CSI36X	Foundation Elective-I (IC)	CSE	2-0-2-0	3	100
7	15CSL37	Analog and Digital Electronics Laboratory	CSE	1-0-2-0	2	100
8	15CSI38	Virtualization Foundations (IC)	CSE	1-0-2-0	2	100
9	15CSH39	Soft Skills Development	CSE	0-2-0-0	1	100
Total				19-2-10-4	26	900

Foundation Elective-I (IC)

Sl. No	Course Code	Course
1	15CSI361	Computer Communication and Networking
2	15CSI362	Creating Interactive and Responsive Web Pages
3	15CSI363	Principles of Programming

IC – Integrated Course **L – Lecture** **T-Tutorials**
P-Practical **S – Self Study**

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY
Department of Computer Science & Engineering
Outcome Based Education (OBE)/Choice Based Credit System (CBCS)
Fourth Semester B.E.- Scheme

Sl. No	Course Code	Course	Teaching Dept.	L-T-P-S (Hrs/week)	Total Credits	Marks
1	15CSM41	Engineering Mathematics-IV (IC)	Mathematics	3-0-2-0	4	100
2	15CST42	Formal Languages and Automata Theory	CSE	3-0-0-0	3	100
3	15CST43	Design and Analysis of Algorithms	CSE	3-0-0-0	3	100
4	15CSI44	Microprocessors (IC)	CSE	3-0-2-0	4	100
5	15CSI45X	Foundation Elective-II (IC)	CSE	3-0-2-0	4	100
6	15CST46X	Engineering Elective-III	CSE	3-0-0-0	3	100
7	15CSL47	Design and Analysis of Algorithms Laboratory	CSE	1-0-2-0	2	100
8	15CSI48	Cloud Computing Foundations (IC)	CSE	1-0-2-0	2	100
9	15CSH49	Soft Skills Development	CSE	0-2-0-0	1	100
Total				20-2-10-0	26	900

Foundation Elective-II (IC)

Sl. No	Course Code	Course
1	15CSI451	UNIX and Shell Programming
2	15CSI452	Object Oriented Programming with C++
3	15CSI453	Introduction to Programming using Python

Engineering Elective-III

Sl. No	Course Code	Course
1	15CST461	Introduction to Cyber Security and Cyber Laws
2	15CST462	Linear Integrated Circuits
3	15CST463	Control Systems

IC – Integrated Course **L – Lecture** **T-Tutorials**
P-Practical **S – Self Study**

SCHEME OF TEACHING AND EXAMINATION

B.E. COMPUTER SCIENCE AND ENGINEERING

V SEMESTER

S. No.	Subject Code	Subject	Teaching Dept.	Teaching Hrs / Week		Examination			
				Theory	Practical	Duration (Hrs)	Marks		
							IA	Exam	Total
1	10IS51	Software Engineering	CSE/ISE	04	-	03	25	100	125
2	10CS52	Systems Software	CSE/ISE	04	-	03	25	100	125
3	10CS53	Operating Systems	CSE/ISE	04	-	03	25	100	125
4	10CS54	Database Management Systems	CSE/ISE	04	-	03	25	100	125
5	10CS55	Computer Networks - I	CSE/ISE	04	-	03	25	100	125
6	10CS56	Formal Languages and Automata Theory	CSE/ISE	04	-	03	25	100	125
7	10CSL57	Database Applications Laboratory	CSE/ISE	-	03	03	25	50	75
8	10CSL58	Systems Software & Operating Systems Laboratory	CSE/ISE	-	03	03	25	50	75
Total				24	06	-	200	700	900

SCHEME OF TEACHING AND EXAMINATION

B.E. COMPUTER SCIENCE AND ENGINEERING

VI SEMESTER

S. No.	Subject Code	Subject	Teaching Dept.	Teaching Hrs / Week		Examination			
				Theory	Practical	Duration (Hrs)	Marks		
							IA	Exam	Total
1	10AL61	Management and Entrepreneurship	CSE/ISE/ MBA	04	-	03	25	100	125
2	10CS62	Unix System Programming	CSE/ISE	04	-	03	25	100	125
3	10CS63/ 10IS662	Compiler Design	CSE/ISE	04	-	03	25	100	125
4	10CS64	Computer Networks - II	CSE/ISE	04	-	03	25	100	125
5	10CS65 / 10IS665	Computer Graphics and Visualization	CSE/ISE	04	-	03	25	100	125
6	10CS66x	Elective I (Group-A)	CSE/ISE	04	-	03	25	100	125
7	10CSL67	Computer Graphics and Visualization Laboratory	CSE/ISE	-	03	03	25	50	75
8	10CSL68	Unix System Programming and Compiler Design Laboratory	CSE/ISE	-	03	03	25	50	75
Total				24	06	-	200	700	900

Elective I – Group A

10CS661/10IS661	Operations Research
10CS662	Signals and Systems
10CS663/10IS663	Data Compression
10CS664/10IS664	Pattern Recognition
10CS665	Stochastic Models and Applications
10CS666/10IS666	Programming Languages

SCHEME OF TEACHING AND EXAMINATION

B.E. COMPUTER SCIENCE AND ENGINEERING

VII SEMESTER

S. No.	Subject Code	Subject	Teaching Dept.	Teaching Hrs / Week		Examination			
				Theory	Practical	Duration (Hrs)	Marks		
							IA	Exam	Total
1	10CS71	Object-Oriented Modeling and Design	CSE/ISE	04	-	03	25	100	125
2	10CS72/ 10IS752	Embedded Computing Systems	CSE/ISE	04	-	03	25	100	125
3	10CS73	Programming the Web	CSE/ISE	04	-	03	25	100	125
4	10CS74	Advanced Computer Architectures	CSE/ISE	04	-	03	25	100	125
5	10CS75x	Elective II (Group-B)	CSE/ISE	04	-	03	25	100	125
6	10CS76x	Elective III(Group-C)	CSE/ISE	04	-	03	25	100	125
7	10CSL77	Networks Laboratory	CSE/ISE	-	03	03	25	50	75
8	10CSL78	Web Programming Laboratory	CSE/ISE	-	03	03	25	50	75
Total				24	06	-	200	700	900

	Elective II – Group B		Elective III – Group C
10CS751/10IS751	Advanced DBMS	10CS761/10IS761	C# Programming and .Net
10CS752	Digital Signal Processing	10CS762/10IS762	Digital Image Processing
10CS753/10IS753	Java and J2EE	10CS763/10IS763	Game Theory
10CS754/10IS754	Multimedia Computing	10CS764/10IS764	Artificial Intelligence
10CS755/10IS74	Data Warehousing and Data Mining	10CS765/10IS765	Storage Area Networks
10CS756/10IS756	Neural Networks	10CS766/10IS766	Fuzzy Logic

SCHEME OF TEACHING AND EXAMINATION

B.E. COMPUTER SCIENCE AND ENGINEERING

VIII SEMESTER

S. No.	Subject Code	Subject	Teaching Dept.	Teaching Hrs / Week		Examination			
				Theory	Practical	Duration	Marks		
							IA	Exam	Total
1	10IS81	Software Architectures	CSE/ISE	04	-	03	25	100	125
2	10CS82	System Modeling and Simulation	CSE/ISE	04	-	03	25	100	125
3	10CS83x	Elective IV(Group-D)	CSE/ISE	04	-	03	25	100	125
4	10CS84x	Elective V(Group-E)	CSE/ISE	04	-	03	25	100	125
5	10CS85	Project Work	CSE		06	03	100	100	200
6	10CS86	Seminar	CSE	-	-	-	50	-	50
Total				16	06		250	500	750

Elective IV – Group D		**Elective V– Group E**	
10CS831/10IS831	Wireless Networks and Mobile Computing	10CS841/10IS841	Ad-hoc Networks
10CS832/10IS832	Web 2.0 and Rich Internet Applications	10CS842	Software Testing
10CS833	VLSI Design and Algorithms	10CS843	ARM Based System Design
10CS834/10IS834	Network Management Systems	10CS844/10IS844	Services Oriented Architecture
10CS835/10IS835	Information and Network Security	10CS845/10IS845	Clouds, Grids and Clusters
10CS836/10IS836	Microcontroller-Based Systems	10CS846	Multi-core Architecture and Programming

NOTE: Students have to register for one Elective from each of the five Elective Group.

Engineering Mathematics-III (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSM31	3:0:2:0	4	CIE:50 SEE:50	3 Hours	BS

Course Objectives:

This course will enable students to :

- The course is aimed at developing the application of mathematical skills in solving the engineering problems using computers.
- Learn to use the partial differential equations in engineering applications.
- Use of Transforms in the engineering problems.
- Able to find the approximated solutions to engineering problems numerically.

Syllabus

Module - I

Partial Differential Equations: Formation of PDE –Eliminating the Arbitrary constants and arbitrary functions, solutions of non homogenous PDE by direct integration, Method of separation of variables. Applications to PDE –Derivation of one dimensional of wave equation and solution by separation of variables-with specified boundary conditions. Derivation of one dimensional of Heat equation and solution by separation of variables-with specified boundary conditions. **08 Hours**

Module - II

Fourier Series: Periodic functions, Dirchlet's conditions, Euler's Formulae-Fourier series of periodic functions of period 2l and 2π, Half range Fourier series, Practical harmonic analysis. **08 Hours**

Module - III

Numerical Methods-I: Numerical solutions of Algebraic and transcendental equations-Regula Falsi Method and Newton Raphson Method. Finite Differences-Forward, Backward and Central differences, Newton's Forward Newton's Backward and Sterling's interpolation formulae. Lagrange's Interpolation formula (without proof). Numerical Differentiation using Newton's Forward and Backward formulae.

08 Hours

Module - IV

Numerical Methods-II: Numerical Integration-Trapezoidal rule, Simpson's $1/3^{rd}$ and $3/8^{th}$ rule. Numerical solutions of ordinary differential equations of first order

and first degree- Picard's method, Taylor's Series method, Modified Eulers Method, Runge-Kutta Method of 4th order and Milne's Predictor-Corrector Method.

08 Hours

Module - V

Introduction to MATLAB, and its family, Menus and toolbars, Types of windows and types of files, MATLAB Help system, Basic calculations in MATLAB, Basic variables, Functions-Elementary Mathematical, Built in and User defined functions. Array operations, Matrix operations, Loops: for and while loops, condition statements- if-then and if-then-else statements, plotting of graphs, working with scripts and files, approximations and errors using MATLAB, solutions of initial value problems, solutions of system of equations. **08 Hours**

List of MATLAB Experiments

Sl.no	Name of Experiment
01	Basics of MATLAB
02	Basic operations in MATLAB
03	Basic Vector operations
04	Basic Matrix operations
	Solution of Linear Equations
	Determination of Eigen values and Eigen vectors of a Square Matrix
	Solution of Linear Equations for Undetermined and Over determined Cases.
05	Basic Operations on Complex Numbers
06	Plotting of 2D and 3D Curves
07	Polynomial Evaluation and Determination of Roots of a Polynomial
	Determination of Polynomial using Method of Least Square Curve Fitting
	Determination of Polynomial Fit , Analyzing Residuals, Exponential fit and Error Bounds from Given Data
08	Use of Functions
09	Differentiation and Integration
10	Solution of linear differential equations
	Numerical Solutions of Ordinary Differential Equations by Euler's Method
	Numerical Solutions of Ordinary Differential Equations by 4th order Runge Kutta Method

Course Outcomes:

On completion of this course, the students are able to :

- Form a partial differential equations and their solutions.
- Expressing the given functions as infinite series of sine and cosine.
- Find approximated solutions by numerical methods.
- Use the MATLAB to solve the various types engineering problems.

Text Books:

1. Dr. B.S. Grewal: "Higher Engineering Mathematics", (Chapters 10, 17, 18, 22, 23, 28-30), Khanna Publishers, New Delhi, 42nd Edition, 2012, ISBN No: 9788174091956.
2. N.P. Bali and Dr. Manish Goyal: "A Text Book of Engineering Mathematics", (Chapters 10,16,17,20,22,23), Laxmi Publications (P) Ltd., New Delhi, 9th Edition, 2014, ISBN: 9788131808320.
3. Rudrapratab: "Getting started with MATLAB", (Chapters 1-4), Oxford University press, United Kingdom, Indian Edition, 2014 (reprinted).

Reference Books:

1. Erwin Kreyszig: "Advanced Engineering Mathematics", (Chapters 11,12,19), Wiley Pvt. Ltd India, New Delhi, 9th Edition, 2011, ISBN 13: 9788126531356.
2. B.V. Ramana: "Higher Engineering Mathematics", (Chapters 17-21,32), Tata Mc Graw – Hill Publishing company Limited, New Delhi, 2nd Reprint, 2007, ISBN 13: 978-0-07063417-0.
3. S.S. Sastry: "Introductory methods of numerical analysis", (Chapters 2,3,6), PHI Learning Private, Delhi, 5th Edition, 2013, ISBN: 978-81-203-4592-8.
4. Stormy Attaway: "A practical introduction to programming and problem solving", Elsevier, Boston, 2nd Edition.

E-Resources:

1. http://bookboon.com/en/essential-engineering-mathematics-ebook
2. https://www.free-ebooks.net/ebook/essential-engineering-mathematics
3. http://www.zums.ac.ir/ebooks/mathematics/essential-engineering-mathemat-ic.
4. https://archive.org/details/AdvancedEngineeringMathematics10thEdition
5. www.mathworks.com

Fundamentals of Computation Engineering

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST32	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the logical notation of fundamental concepts such as sets, relations and functions.
- Understand the syntax and semantics of propositional and predicate logic.
- Translate statements from a natural language into its symbolic structures in logic.
- Understand the basic concepts of graph theory.
- Learn how to use graphs as a powerful modeling tool to solve practical problems in various fields.
- Get familiarized with modeling of computational methods.

Module - I

Set Theory: Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams.
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic. **06 Hours**

Module - II

Fundamentals of Logic contd.: Logical Implication – Rules of Inference, The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems. **06 Hours**

Module – III

Relations and Functions: Cartesian Products and Relations, Functions –Plain and One-to-One, Onto Functions –Special Functions, The Pigeon-hole Principle, Function Composition and Inverse Functions. Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **10 Hours**

Module – IV

Introduction to Graph Theory: Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, , Hamilton Paths and Cycles. Graph Coloring, and Chromatic Polynomials.
Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes. **10 Hours**

Module – V

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, combinations – The Binomial Theorem, Combinations with Repetition.

The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle. **08 Hours**

Course outcomes:

On completion of this course, the students are able to :

- Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations and functions.
- Develop the syntax and semantics of propositional and predicate logic.
- Define, compare and recognize relations and functions and Identify relations and functions with graphs, tables and sets of ordered pairs.
- Apply the abstract concepts of graph theory in modeling and solving non-trivial problems in different fields of study.
- Demonstrate the ability to solve problems using counting techniques and combinatorics.

Text Book:

1. Ralph P. Grimaldi: "Discrete and Combinatorial Mathematics", (Chapters 1-3, 5,7,8,11,12), Pearson Education, 5th Edition, 2006, ISBN: 8177584243, 9788177584240.

Reference Books:

1. Kenneth H. Rosen: "Discrete Mathematics and its Applications", McGraw Hill, New Delhi, 7th Edition, 2010, ISBN : 0073383090.
2. J K Sharma: "Discrete Mathematics", Trinity, India, 4th Edition, 2015, ISBN: 978-93-5138-143-3.
3. D.S. Chandrasekharaiah: "Graph Theory and Combinatorics", Prism, Bengaluru, 4th Edition, 2013, ISBN: 978-81-7286-698-3.
4. Richard A. Brualdi: "Introductory Combinatorics", Pearson Education, India, 4th Edition, 2004, ISBN: 978-0-13-602040-0.

E-Resources:

1. https://www.pearsoned.co.in/grimaldidcm5e

Data Structures with C (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI33	3:0:2:4	5	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basics of structures and pointers.
- Learn the data structure for a specified application.
- Understand the impact of data structures on the performance of programs.
- Get exposed to various data structure concepts, such as stacks, queues, linear lists, heaps, trees and graphs.

Syllabus

Module - I

Structures and Unions: Introduction, Defining a structure, Declaring structure variables, Accessing structure members, Structure initialization, Size of structures, Copying and comparing structure variables, Array of structures, Arrays within structures, Structures within structures, Structures and functions, Unions.

Pointers: Introduction, Accessing the variable and its address through the pointer, Declaring and initialization of pointer variables, Chain of pointers, Pointer expressions, Pointer increments and Scale factor, Array of pointers, Pointer as function arguments, Functions returning pointers, Pointers to functions, Pointers and arrays, Pointers and character strings, Pointers and structures. Dynamic memory allocation - Introduction, malloc(), calloc(), realloc(). **08 Hours**

Module - II

Stacks: Definition, Representing stacks in C - Implementing the PUSH and POP operation with example and testing for exceptional conditions, Infix, Postfix, and Prefix expressions. Program to evaluate a postfix expression with example, Program to convert an expression from infix to postfix. Recursion: Definition, Recursion in C with examples.

Queues: The Queue and its sequential representation, Implementation of queues in C, Insert and delete operations, Priority queue, Array implementation of a priority queue. **08 Hours**

Module - III

Linked Lists: Introduction, Inserting and removing nodes from a list, Linked list implementation of stacks, queues and priority queues, List operations, get node and

free node. Lists in C - Array implementation of lists and its limitations, Allocating and freeing of dynamic variables, Linked lists using dynamic variables and compare with array implementation of lists, Non-integer and non-homogeneous lists, Circular lists: Primitive operations on circular lists, implement stack and queue as a circular list, Doubly linked lists: Inserting and removing nodes from a list, Addition of long integer using Doubly linked lists. **08 Hours**

Module - IV

Trees: Introduction, Binary trees, Binary tree traversals: preorder, inorder and postorder, Threaded binary trees : insertion and deletion of a node, Heaps: insertion and deletion of a node, Max heap and min heap, Binary search trees: insertion and deletion of a node, searching a key element in binary search tree, Selection trees, Forests, Representation of disjoint sets, Counting binary trees. **08 Hours**

Module - V

Efficient Binary Search Trees: Optimal binary search trees, AVL trees: insertion and deletion of node, Red-black trees: insertion and deletion of a node, Splay trees: insertion and deletion of a node, Graph: Introduction, Abstract data type of graph. **08 Hours**

Laboratory

Design, develop and implement the specified algorithms for the following problems using C/C++ language in LINUX / Windows environment and simulate the programs by using virtual lab.

1. Design, develop and execute a program in C based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_Name (a string of characters), Basic_Salary (an integer), All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic, Income Tax (IT) = 30% of the gross salary (gross salary = Basic_Salary + All_Allowance), Net_Salary = Basic_Salary + All_Allowances – IT).
2. Using circular representation for a polynomial, design, develop and execute a program in C to accept two polynomials, add them and then print the resulting polynomial.
3. Design, develop and execute a program in C to simulate the working of a Stack of integers using an array. Provide the following operations:

   a. Push b. Pop c. Display

4. Design, develop and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations:

   a. Insert b. Delete c. Display

5. Design, develop and execute a program in C to convert a given valid parenthesized infix arithmetic expression to postfix expression and then to print both the expressions. The expression consists of single character operands and the binary operators +, -, * and /.

6. a. Design, develop and execute a program in C to evaluate a valid postfix expression using Stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are +, -, * and /.

   b. Simulation of the above program by using virtual lab.

7. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front of the list as well as to delete an element from the front of the list (FIFO), display the contents of the list.

8. a. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front of the list as well as to delete an element from the rear of the list or vice versa (LIFO), display the contents of the list.

   b. Simulate a program to use Linked list to do operations on polynomials. (virtual lab)

9. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front/rear end of the list and search and delete a key element of the list, if exists and display the contents of the list.

10. Design, develop and execute a program in C to create a max heap of integers by accepting one element at a time and by inserting it immediately in to the heap. Use the array representation for the heap. Display the array at the end of insertion phase.

11. Write a C program to support the following operations on a doubly linked list where each node consists of integers,

    a. Create a doubly linked list by adding each node at the front.

    b. Insert a new node to the left of the node whose key value is read as an input.

    c. Delete the node of a given data, if it is found, otherwise display appropriate message.

    d. Display the contents of the list.

12. Design, develop and execute a program in C to implement BIN_TREE that represents a Binary Tree, with functions to perform inorder, preorder and postorder traversals and demonstrate the traversals.

Course Outcomes:

On completion of this course, the students are able to :

- Describe the concepts of structures and pointers.
- Explain and illustrate the various stack and queue operations.
- Interpret and design the programs using linked lists.
- Implement binary trees, threaded binary trees, binary search trees using C language.
- Implement AVL trees, Red-black trees and Splay trees using C language.

Text Book:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed : "Fundamentals of Data Structures in C", (Chapters 1-6,9,10), Universities Press-India, 2nd Edition, 2008, ISBN-13: 978-8173716058.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI Learning Private Limited-India, 3rd Edition, 2010, ISBN-13: 978-8120340077.
2. Aaron M. Tenenbaum: "Data Structures using C", Pearson Education, India, 2nd Edition, 2003, ISBN-13: 978-8131702291.

E-Resources:

1. https://www.cs.princeton.edu/
2. https://www.opendatastructures.org/ods-cpp
3. https://www.lib.mdp.ac.in/ebook/DSa
4 https://ww.cs-fundamentals.com/data-structures/introduction-to-data-structures.php
5. https://www.cprogramming.com/algorithms-and-data-structures.html
6. https://www.cprograms.in/

Analog and Digital Electronics

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST34	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand applications of diode as clippers, Limiters and clampers.
- Know the different types of amplifiers constructed from BJTs and MOSFETs.
- Acquire the knowledge about Boolean laws and theorems.
- Understand the various types of data processing circuits.
- Learn the working principles of various Flip-Flops, registers, counters.
- Understand the basic concepts of HDL.

Syllabus

Module - I

Diode Circuits: The ideal diode, The second approximation, The third approximation, DC resistance of a diode, Load lines, Clippers and Limiters, Clampers.

BJT Amplifiers: Emitter-biased amplifier, Small-signal operation, AC Beta, AC resistance of the emitter diode, Two- transistor models, Analyzing an amplifier, Voltage gain. **07 Hours**

Module - II

MOSFETs: The Depletion-mode MOSFET, D-MOSFET curves, Depletion-mode MOSFET amplifiers, The enhancement-mode MOSFET, CMOS, E-MOSFET amplifiers.

OP-AMP's in Waveform conversion and Generation Circuits: Sine to rectangular, Rectangular to triangular, Triangular to pulse conversion circuits, Relaxation oscillator, Generating triangular waves. **09 Hours**

Module - III

Combinational Logic Circuits: Boolean laws and theorems, Sum-of-Products method, Truth table to Karnaugh map, Pairs Quads, and Octets, Karnaugh simplifications, Don't-care conditions, Product-of-Sums method, Product-of-Sums simplifications, Simplification by Quine-McClusky method, Introduction to HDL, HDL implementation models.

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, Encoders, HDL implementation of data processing circuits. **09 Hours**

Module - IV

Flip-Flops: Gated FLIP-FLOPs, Edge-triggered D FLIP-FLOP, Edge-triggered JK FLIP-FLOP, JK Master-slave FLIP-FLOP, Various representation of FLIP-FLOPs, HDL implementation of FLIP-FLOP.

Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out. **08 Hours**

Module - V

Registers: Applications of shift registers, Register implementation in HDL .

Counters: Asynchronous counters, Decoding gates, Synchronous counters, Changing the counter modulus, Decade counters, Counter design as a synthesis problem.

07 Hours

Course Outcomes:

On completion of this course, the students are able to :

- Use diodes in clippers, limiters and clampers.
- Analyze and design transistor and MOSFET amplifiers in different configurations.
- Apply K-Map and Quine-McClusky methods to simplify the given Boolean expressions.
- Design and implement registers using Flip-Flops.
- Design and develop counters using Flip-Flops.

Text Books :

1. Albert Malvino, David Bates: "Electronic Principles", (Chapters 3,4,8,12,20), TMH, New Delhi, 8th Edition, 2015, ISBN-9780073373881.
2. Donald P Leach, Albert Paul Malvino and Goutam Saha: "Digital Principles and Applications", (Chapters 3,4,8- 10), Tata McGraw Hill, New Delhi, India, 8th Edition, 2014, ISBN: 9789339203402.

Reference Books :

1. Robert L. Boylestad, Louis Nashelsky: "Electronic Devices and Circuit Theory", PHI/Pearson Education, New Delhi, 10th Edition, 2012, ISBN: 9788131764596.
2. David A. Bell: "Electronic Devices and Circuits", Oxford University Press, New Delhi, India, 5th Edition, 2010, ISBN: 9780195693409.
3. M Morris Mano: "Digital Logic and Computer Design", Pearson Education, Prentice Hall, 11th Edition, 2009, ISBN: 9788177584097.
4. R D Sudhaker Samuel: "Illustrative Approach to Logic Design", Sanguine-Pearson, New Delhi India, 2012, ISBN: 9788131765081.

E-Resources:

1. https://www.talkingelectronics.com/download/Malvino_electronic_principles.pdf.
2. https://www.rtna.ac.th/departments/elect/data/EE304/Electronic%20Devices%20and%20Circuit%20Theory.pdf
3. https://www.abebook.com/Digital_Principles_Applications_Seventh_Edition_Albert/4893172428/bd

Computer Organization

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST35	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Become familiar with the basics of computer structure and its performance.
- Learn basics of assembly language programming.
- Understand connections of peripheral devices and interrupts generated by them.
- Understand the basics of data communication by using different types of buses.
- Learn internal structure of memory and CPU.

Syllabus

Module - I

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Performance -Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement, Historical Perspective.

08 Hours

Module - II

Machine Instructions and Programs: Numbers, Arithmetic operations and Characters, Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions. **09 Hours**

Module - III

Input/ Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB. **08 Hours**

Module - IV

Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories. **08 Hours**

Module - V

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand multiplications, Fast Multiplications, Integer Division, Floating-point Numbers and Operations. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Describe basic structure and performance of computer.
- Explain basics of assembly language programming.
- Explain connections of peripheral devices, interrupts and DMA operations.
- Use different types of buses for data transmission and design of basic types of memory.
- Construct different types of arithmetic circuits.

Text Book:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: "Computer Organization", (Chapters 1,2,4-6) TMH, Mcgraw-Hill College Division, 5th Edition, 2002. ISBN :9781259005275.

Reference Books:

1. William Stallings: "Computer Organization and Architecture", PHI, Pearson Education, Delhi, 10th Edition, 2016, ISBN: 9780134101613.
2. David. A. Patterson, John L. Hennessy: "Computer Organization and Design – The Hardware / Software Interface", ARM Edition, 5th Edition, Elsevier, 2014, ISBN: 97801240776263.

E-Resources:

1. https://books.google.co.in/books?isbn=0071089004
2. https://books.google.co.in/books?isbn=8177589938
3. https://books.google.co.in/books?isbn=0124078869

Computer Communication and Networking (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI361	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Understand the basics of data communication system and network models.
- Identify the need and techniques for digital and analog transmissions.
- Get exposed to different error detection and correction methods.

Syllabus

Module - I

Introduction to Communications: Data Communications, Networks - Physical structure, Categories of Networks, Network Models –The OSI Model, TCP/IP Protocol Suite. **06 Hours**

Module - II

Digital Transmission: Digital to Digital Conversion – Line coding, Line coding schemes (Unipolar, Polar, Bipolar only), Analog to Digital Conversion – PCM Encoder and Decoder, Transmission Modes. **05 Hours**

Module - III

Analog Transmission: Digital to Analog conversion - Introduction, ASK, PSK, FSK, Analog to Analog conversion – Introduction, AM, PM, FM. **05 Hours**

Module - IV

Multiplexing and Spectrum Spreading: Multiplexing - Introduction, FDM, Synchronous TDM, Statistical TDM, , Spread Spectrum – Introduction, FHSS, DSSS. **05 Hours**

Module - V

Error Detection and Correction: Introduction, Block Coding, Cyclic Codes – CRC, Polynomials, Cyclic code encoder using Polynomials, Advantages of cyclic codes. Checksum. **05 Hours**

Laboratory

1. Write a program to convert digital to analog data transmission.
2. Write a program to convert analog to digital data transmission.

3. Write a program for error detecting code using CRC-CCITT (16 bits).
4. Using TCP/IP sockets write a client server program to make the client send the file name and to make the server send back the contents of the requested file if present.

Course Outcomes:

On completion of this course, the students are able to :

- Describe the basics of data communication system and network models.
- Distinguish between different techniques of digital transmissions.
- Compare different methods of analog transmissions.
- Explain various types of multiplexing and spread spectrum mechanisms.
- Solve problems of error detection and correction using Block coding and CRC mechanisms.

Text Book:

1. Behrouz A. Forouzan: "Data Communication and Networking", (Chapters 1,2, 4-6, 10), McGraw Hill Education, New Delhi, India, 5th Edition, Copyright: 2013. Publication Date: February 17, 2012, ISBN: 9781259064753.

Reference Books:

1. William Stallings: "Data and Computer Communication", Pearson Education, Delhi, 8th Edition, 2007, ISBN: 9788131715369.
2. Craig Zacker: "The Complete Reference Networking", McGraw-Hill Education, New Delhi, India, 1st Edition, 2002, ISBN: 13: 978-0070474161.
3. Wayne Tomasi: "Introduction to Data Communications and Networking", Pearson Education, Delhi, 1st Edition, ISBN 13: 978-8131709306.

E-Resources:

1. http://www.mhhe.com/engcs/compsci/forouzan/frontmatter.pdf.
2. http://ebookinga.com/data-communication-and-networking-tata-mcgraw-hil.
3. https://www.goodreads.com/book/show/209441.Introduction_to_Data_Communications_and_Networking.

Creating Interactive and Responsive Web Pages (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI362	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Learn the evolution of the World Wide Web and its relevance in today's world.
- Get a clear understanding of the technologies involved in developing a web-site.
- Learn web technology with a focus on creating interactive and responsive web pages.

Syllabus

Module - I

HTML 5 and CSS: Introduction to Hyper Text Markup Language, Key components of HTML document, HTML elements, Headers, Linking, Images, Unordered Lists, and Nested and ordered Lists.

Tables, Divs and forms: HTML Tables and Formatting, HTML Forms, Internal Linking, Creating and Using Images, Maps, Div and span tags. Introduction CSS, CSS selector, positioning, layouts, debugging. **06 Hours**

Module - II

JavaScript: Browser and Document object, scripts and HTML Document, variables, expressions, Data type conversions, decisions and loops, control structure, windows Document object, forms and form handling elements, scripting, event handling.

05 Hours

Module - III

jQuery: Using selectors with jQuery, Manipulating page elements with jQuery, jQuery event model, jQuery and Ajax, jQuery animation and advanced effects, jQuery plugins. **05 Hours**

Module - IV

Bootstrap: Bootstrap Scaffolding, Bootstrap CSS, Bootstrap Layout Components, Bootstrap JavaScript Plugins, Using Bootstrap. **05 Hours**

Module - V

XML: What is XML? What are the differences between HTML and XML, what is the purpose of XML?

AJAX: AJAX Introduction, AJAX XML Http, AJAX Request, AJAX Response, AJAX with Server side. **05 Hours**

Hands on

1. HTML and CSS -

a. Create a HTML page to display the following content <Ensure the format is same as shwn below>; Use HTML Tables UnorderList and OrderList (UL and OL):

1. Development Environment • Eclipse (SpringSource Tool Suite distribution) • Apache Tomcat/VMware® vFabric™ tc Server • Spring Insight • Testing tools	**2. Spring Overview** • Introduction to Spring configuration • Bean life cycle • Simplifying configuration • Integration testing with Spring
3. Getting Started with Spring Web MVC • Spring model-view-controller (MVC) overview • DispatcherServlet • Controller programming model overview • Spring MVC views • Simplifying configuration	**4. Spring MVC Configuration Options** • Spring MVC infrastructure Beans • URL mappings • Handler interceptors and handler adapters • Exception resolvers • Message source

b. Create a HTML page to display testimonials received from customers along with their picture and Name as shown below:

2. Java Script and jQuery

a. jQuery form validations:

Front-End: Develop below form using HTML to create new user:

Field Details:

1. Name <String, Length(16), Mandatory, Validations: Minimum Length: 3, Can accept special characters>
2. Email <String, Length(60), Mandatory, Validations: Should be a valid email id>
3. Password <String, Length(16), Mandatory, Validations: Minimum Length: 5, Can accept special characters>

- Create Buttons - "Create an account" and "Cancel"
- Form should be Scrollable
- For field validations, use jQuery
- Ensure all validations pertaining to Name, Email and Password are taken care. If the use enters incorrect values appropriate error message should be displayed and should allow the user to enter correct data

b. jQuery image slider:

In an HTML page, insert a minimum of 5 images; Ensure inserted images are scrollable.
Hint: To make images scrollable use jQuery image slider or use javascript.

Sample screen shot :

3. Develop below form using HTML to Search and Book Tickets:

Validations :

1. All fields are Mandatory except “Single Lady” Field
2. Onwards date must be less than Return date

4. Bootstrap, AJAX and jQuery:

a. Create a Bootstrap Page that helps maintain Employee Information in an organization.

b. When clicked on the “Add New Employee” button, load a dialog box as shown below

Validations:

1. All fields are Mandatory.
2. On successful submission of the form, the new employee details has to be appended as a last row in the table.
3. When clicked on the Edit icon, a similar pop-up as the "Add new Employee" form has to be displayed with the input fields populated with appropriate values. When the form is submitted in the Edit flow, ensure all the validations are in place. The name of the button in the Edit flow has to be "Update" instead of "Add".
4. When clicked on the Delete icon, a confirmation dialog box has to be displayed with a message "Are you sure, you want to delete this entry?" If the user clicks "Yes", the corresponding row has to be deleted from the table. If the user clicks "No" the table has to remain unaffected.

Course Outcomes:

On completion of this course, the students are able to :

- Develop web layouts with style sheets and web screens in a presentable form.
- Write interactive web pages through form validations and other methods. Use the same in UI development.
- Use the Java Script libraries to accelerate UI development.
- Design and develop responsive and mobile first web pages.
- Develop applications by using synchronous and asynchronous communication over web.

Text Book:

1. Jon Duckett: "Web Design with HTML, CSS, JavaScript and jQuery Set", Wiley, 1st Edition, 2014, ISBN 13: 978-1118907443.

Reference Books:

1. Jake Spurlock: "Bootstrap, Shroff", O'Reilly Media, United States of America, 1st Edition, 2013, ISBN: 978 -1 -4493-4391-0.
2. Bear Bibeault, Yehuda Katz and Aurelio De Rosa: "jQuery in Action", Dreamtech Press, New Delhi, India, 3rd Edition, 2015, ISBN: 978-1617292071.

E-Resources:

1. http://www.w3schools.com/
2. https://learn.jquery.com/
3. https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basics

Principles of Programming (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI363	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Learn data, data types, and basic statements.
- Gain knowledge about call-return architecture and ways of implementing them.
- Understand object-orientation, concurrency, and event handling in programming languages.
- Develop programs in non-procedural programming paradigms.

Syllabus

Module - I

Data, Data Types, and Basic Statements-I: Names, Variables, Binding, Type checking, Scope, Scope rules, Lifetime and garbage collection, Primitive data types, Strings, Array types, Associative arrays, Record types, Union types, Pointers and references. **05 Hours**

Module - II

Data, Data Types, and Basic Statements-II: Arithmetic expressions, Overloaded operators, Type conversions, Relational and boolean expressions, Assignment statements, Mixed-mode assignments, Control structures, Selection, Iterations, Branching, Guarded statements. **05 Hours**

Module - III

Subprograms and Implementations-I: Subprograms, Design issues, Local referencing, Parameter passing, Overloaded methods, Generic methods, Design issues for functions. **05 Hours**

Module - IV

Subprograms and Implementations-II: Semantics of call and return, Implementing simple subprograms, Stack and dynamic local variables, Nested subprograms, Blocks, Dynamic scoping. **05 Hours**

Module - V

Object-Orientation, Concurrency, and Event Handling: Object-orientation, Design issues for OOP languages, Implementation of object-oriented constructs, Concurrency, Exception handling. **06 Hours**

Laboratory

1. Implement all major functions of string.h in single C program using switch case to select specific function from user choice (like strlen, strcat, strcpy, strcmp, strrev).
2. Write a program (WAP) in C to reverse a linked list iterative and recursive.
3. WAP in C to implement iterative Towers of Hanoi.
4. WAP in C++ to count the no's of object of a class with the help of static data member, function and constructor.
5. WAP in C++ to declare a class Time with data members mm for minutes, ss for seconds and hh for Hours.
6. Define a parameterize constructor to assign time to its objects. Add two time objects using member function and assign to third objects. Implement all possible cases of time.

Course Outcomes:

On completion of this course, the students are able to :

- Describe data and data types of programming languages.
- Explain basic statements of programming languages.
- Design subprogram constructs.
- Implement simple subprogram.
- Apply object-oriented, concurrency, and event handling programming constructs.

Text Book:

1. Robert W. Sebesta: "Concepts of Programming Languages", (Chapters 5-10,12, 14), 10th Edition, Addison Wesley, 2012, ISBN: 0-13-607347-6.

Reference Books:

1. R. Kent Dybrig:"The Scheme programming language", 4th Edition, MIT Press, 2009, ISBN: 978-0-262-51298-5.
2. Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009, ISBN: 0 262 15039 5.
3. Michael L. Scott, "Programming Language Pragmatics", 3rd Edition, Morgan Kaufmann, 2009, ISBN-13: 978-0123745149, ISBN-10: 0123745144.

E-Resources:

1. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=robert+w.+sebesta+concepts+of+programming+languages
2. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=michael+l.+scott+programming+language+pragmatics
3. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=programming+language+pragmatics+morgan+kaufmann+pdf

Analog and Digital Electronics Laboratory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSL37	1-0-2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand and design diode circuits such as clippers and clampers.
- Learn the design of OP-AMP circuits.
- Learn how to implement combinational circuits.
- Know how to design and implement sequential circuits.

Part-A

1. Design and construct a suitable circuit and demonstrate the working of positive clipper, and clamper using diodes.
2. Construct a suitable circuit and determine the frequency response and bandwidth of a CE amplifier.
3. Design and implement relaxation circuit using OP-AMP to generate a rectangular wave from for a given frequency.
4. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
5. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop and demonstrate its working.
6. Design and implement a ring counter using 4-bit shift register and demonstrate its working.
7. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate its working.

Part-B (Virtual Lab)

1. Demonstrate the working of the positive clipper and clamper circuits.
2. Build the CE amplifier circuit and determine the voltage gain for two different values of supply voltage and for two different values of emitter resistance.
3. Build CMOS inverter using a simulation package and verify its truth table.
4. Develop the Verilog / VHDL code for an 8:1 multiplexer. Simulate and verify its working.
5. Develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
6. Develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.
7. Develop the Verilog / VHDL code for switched tail counter. Simulate and verify its working.

Course Outcomes:

On completion of this course, the students are able to :

- Analyze and design transistor and MOSFET Amplifiers in different configurations.
- Construct various diode circuits and OP-AMP circuits.
- Realize and verify truth tables of D Flip-Flop and 8:1 Multiplexer.
- Design and implement ring counter.
- Design and implement synchronous and asynchronous counters.

Text Books:

1. Albert Malvino, David Bates: "Electronic Principles", 8th Edition, TMH, New Delhi, 2015, ISBN: 9780073373881.
2. Donald P Leach, Albert Paul Malvino and Goutam Saha: "Digital Principles and Applications", 8th Edition, Tata McGraw Hill, New Delhi, India, 2014, ISBN: 9789339203402.

E-Resources:

1. http://elearning.vtu.ac.in>E-Notes>LD manual.
2. http://Faqexplorer.com

Virtualization Foundations (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI38	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Get a strong foundation for mastering computer virtualization.

Syllabus

Module - I

Introduction to Virtualization: Definition of Virtualization, Traditional versus Virtualized System, How Virtualization works, Need for Virtualization, Benefits of Virtualization, Definition of Hypervisor, Type-1 Hypervisor, Type-2 Hypervisor, Comparison of Type1 and Type2 Hypervisors, Types of Virtualization - Server Virtualization, Desktop Virtualization, Application Virtualization, Network Virtualization, Storage Virtualization. **03 Hours**

Module - II

VMware Player: Overview, Introduction to VMware Player, System Requirements, Installation of VMware Player, Enabling VT-x/AMD-v in BIOS, Creating a blank Virtual Machine, Installation of Open Suse 64-bit, Introduction to Virtual Appliance, Demonstration of Virtual Appliance, Introduction to Unity mode, Demonstration of Unity mode. **03 Hours**

Module - III

ESXi: Introduction to ESXi, ESXi Architecture, System Requirements, Installing ESXi, Installing Windows 2008 R2, Installing vSphere Client, Creating a VM on ESXi using vSphere Client, Installing a Guest OS on ESXi. **02 Hours**

Module - IV

Networking and Storage: Introduction to Virtual Networking, Virtual Networking - Components and Concepts, Virtual Networking in ESXi, Introduction to Storage, How Virtual Machines Access Storage, Types of Physical Storage. **03 Hours**

Module - V

VMware Converter: Overview, Installation of VMware Converter, Converting a Physical Computer to a VM, Powering-on the converted VM. **02 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Explain what virtualization is, and get an in-depth understanding of how things work at the ring level.
- Install VMware Player and create virtual machines using VMware Player. In specific, the module will guide on setting up a virtual lab environment using VMware Player.
- Install VMware ESXi, working with the vSphere Client, and managing a host using the vSphere Client.
- Cover the concepts of Virtual Networking and Storage with reference to ESXi.
- Use VMware Converter to convert physical computer into a virtual machine.

Reference Books:

1. Matthew Portnoy: "Virtualization Essentials", Wiley, 2012, ISBN: 978-1-118-17671-9.
2. Nelson Ruest, Danielle Ruest: "Virtualization, A Beginner's Guide", McGraw-Hill Education, 2009, ISBN-13: 978-0071614016.

E-Resources:

1. http://www.vmware.com/files/pdf/gated-vmw-ebook_virtualization-essentials.pdf.

Soft Skills Development

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSH39	0:2:0:0	1	CIE:50 SEE:50	2 Hours	HSS

Course Objectives:

This course will enable students to :

- Improve the communication skills.
- Develop the art of presentation and writing effectively.
- Enhance the technical knowledge.

To improve the communication and presentation skills, every student has to give a seminar on technical topics assigned by the supervisors. Each course coordinator/ faculty members will be assigned with few students to guide and monitor the presentation. The presentation shall be for 15 minutes. A brief report on the seminar has to be submitted by the student to the concerned department after completion of the seminar. The report shall be signed by the supervisor and the Head of the concerned department.

The objective of the seminar is to introduce students to the major constituent of technology that is concerned with critically reading, understanding, summarizing, explaining and presenting existing technical topics. Students have to refer one or more topics that are assigned to them by their supervisors. The idea behind the seminar system is to familiarize student more extensively with the methodology of their chosen subject, allow them to develop presentation skills and also interact with example of practical problems.

Course Outcomes:

On completion of this course, the students are able to :

1. Get rid of stage fear and answer questions from audience.
2. Communicate confidently and fluently.
3. Comprehend and prepare reports effectively.

Engineering Mathematics-IV (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSM41	3:0:2:0	4	CIE:50 SEE:50	3 Hours	BS

Course Objectives:

This course will enable students to :

- The course is aimed at developing the application of mathematical skills in solving statistics and probability problems using computers.
- Applying the sample analysis on the large data.
- Learn to analyze the different Queuing Models.
- Introduction of Statistical Software's.

Syllabus

Module - I

Random Variables: Discrete probability distribution, Continuous probability distribution, Expectation, Variance, Probability generating function, Binomial distribution, Poisson distribution, Exponential distribution and Normal distribution.

08 Hours

Module - II

Joint Probability: Joint probability distribution, discrete and independent random variables, expectation, covariance, correlation coefficient, probability vectors, stochastic matrices, fixed point matrices, regular stochastic matrices, Markov chains, higher transition probabilities, stationary distribution of regular Markov chains and absorbing states. **08 Hours**

Module - III

Sampling: Sampling and inference: Sampling distribution, testing of hypothesis, level of significance, confidence limits. Test of significance of large samples, sampling of variables, central limit theorem, confidence limits for unknown means, students t-distribution.

Queuing Theory: Terminology of Queuing system, Queuing models, Simulation.

08 Hours

Module - IV

Information Theory and Coding: Introduction, Entropy, Joint Entropy and Conditional Entropy, Relative Entropy and Mutual Information, Relationship Between Entropy and Mutual Information, Chain Rules for Entropy, Relative Entropy and Mutual Information, Jensen's Inequality and its Consequences, Log Sum Inequality and its Applications, Data-Processing Inequality, Sufficient Statistics, Fano's Inequality.

08 Hours

Module - V

Introduction to R, Basic Data types, vector operations, matrix construction, lists, data frames, Elementary statistics with R-Qualitative and Quantitative data, Numerical measures, probability distribution, interval estimation and simple linear regression.

08 Hours

List of Lab Experiments

Sl.No	Name of the Experiment
1	Introduction to R Software and basic commands
2	Demonstration and operations of Vectors
3	Operations of Matrices
4	Demonstration of Lists
5	Demonstration of Data Frames
6	Qualitative Data Analysis
7	Quantitative Data Analysis
8	Numerical Measures of Data
9	Probability Distribution
10	Linear Regressions

Course Outcomes:

On completion of this course, the students are able to :

- Find the probability using different distributions.
- Apply the stochastic process and Markov chain in prediction of future events.
- Calculate the various parameters of Queuing models.
- Analyze the data by using the concepts of sampling theory.
- Use the statistical software R for analyzing the data .

Text Books:

1. Dr. B.S. Grewal: "Higher Engineering Mathematics", Khanna publishers, New Delhi, 42nd Edition, 2012, ISBN: 9788174091956.

2. R.E.Walpole, R.H.Myers.R.S, L.Myers and K.Ye: "Probability and Statistics for Engineers and Scientists", Pearson Education, Delhi, 8th Edition, ISBN 13: 9780131877115.
3. W.N.Venables, D.M.Smith : "An introduction to R. R- manual".

Reference Books:

1. Erwin Kreyszig: "Advanced Engineering Mathematics", (Chapters 13,14,19,21,24,25), Wiley Pvt. Ltd., India, New Delhi, 9th Edition, 2011, ISBN 13: 9788126531356.
2. John Verzani: "Using R for introductory Statistics", Champan and Hall/ CRC, New York, Washington D.C., ISBN: 978-1-59327-384-2.
3. Sheldom M Ross: "Probability models for Computer Science", Academic Press, 2009, ISBN: 9780124079489.
4. Murray R Spiegel, John Schiler and Alu Srinivasan: "Probability and Statistics", Schaum's Outline series, 2nd Edition, ISBN: 9780071795579.

E-Resources:

1. http://www.zums.ac.ir/ebooks/mathematics/essential-engineering-mathematic.
2. https://archive.org/details/AdvancedEngineeringMathematics10thEdition
3. https://www.r-project.org/
4. www.r-tutor.com

Formal Languages and Automata Theory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST42	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basic concepts of finite automata.
- Learn the regular expressions, languages and their applications.
- Describe the context free grammars, languages and their applications.
- Acquire knowledge about push down automata, languages and its types.
- Introduce Turing machines.

Syllabus

Module - I

Introduction to Finite Automata: Introduction to Finite Automata, The central concepts of Automata theory, Deterministic finite automata, Nondeterministic finite automata. **08 Hours**

Module - II

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. **08 Hours**

Module - III

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. **08 Hours**

Module - IV

Context-Free Grammars and Languages: Context free grammars, Parse trees, Applications, Ambiguity in grammars and Languages.

Properties of Context-Free Languages: Normal forms for CFG's, Closure properties of CFL's. **08 Hours**

Module - V

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

Introduction to Turing Machine: The Turing machine. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Design the different types of automata for given regular expression and regular languages.
- Write the grammar for the given regular expressions.
- Draw the parse tree for the given context free grammars.
- Design the PDA for the given languages.
- Convert the given PDA to its equivalent CFG and vice versa.

Text Book:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: "Introduction to Automata Theory, Languages and Computation", (Chapters 1-8), Pearson New International Edition, 3rd Edition, 2014, ISBN-13: 978-0321455369.

Reference Books:

1. Peter Linz: "An Introduction to Formal Languages And Automata", Jones and Bartlett Publishers, 5th Revised Edition, 2011, ISBN-13: 978-1449615529, ISBN-10: 144961552X.
2. K.L.P. Mishra: "Theory of Computer Science, Automata, Languages, and Computation", PHI Learning, 3rd Edition, 2009, ISBN 10: 8120329686 ISBN 13: 9788120329683.

E-Resources:

1. http://www.ebook777.com/theory-finite-automata-introduction-format-languages.
2. http://www.techmela.ucoz.com/_id/o/22-introuctionto.pdf

Design and Analysis of Algorithms

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST43	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Acquire the knowledge of Algorithm and problem solving technique.
- Learn how to analyze the complexity of an algorithm in terms of time and space.
- Understand different techniques like divide and conquer, decrease and conquer etc., to solve problems.

Syllabus

Module - I

Introduction: What is an algorithm? Fundamentals of algorithmic problem solving, Fundamentals of the analysis of algorithm efficiency, Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms Brute Force Approaches: Introduction, Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching. **08 Hours**

Module - II

Divide and conquer: Divide and Conquer: General Method, Binary Search, Merge Sort, Quick Sort and its performance.

The greedy method: The General Method, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Single Source Shortest Paths. **08 Hours**

Module - III

Decrease and conquer approaches: Introduction, Insertion Sort, Depth First Search and Breadth First Search, Topological Sorting.

Transfer and conquer: Introduction, Balanced search trees, Heap and Heap sort.

08 Hours

Module - IV

Space-Time Trade-offs: Introduction, Sorting by Counting, Input Enhancement in String Matching (Horspool algorithm).

Dynamic programming: The General Method, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, The Travelling Salesperson problem, Computing a Binomial co-efficient. **08 Hours**

Module - V

Limitations of algorithmic power and coping with them: Lower-Bound Arguments, Decision Trees.

Backtracking: n - Queens problem, Subset – Sum Problem.

Hashing: Introduction, Open hashing, Closed hashing.

Branch and bound: Assignment problem, Knapsack problem. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Identify asymptotic notations and basic efficiency classes.
- Solve problems using various techniques like greedy and divide-and-conquer.
- Compute problems using various techniques like decrease-and-conquer and transfer-and-conquer.
- Use different algorithms like TSP, Floyd's etc. to solve real world problems.
- Develop solutions for n - Queens problem, Subset – Sum Problem, Assignment problem, Knapsack problem etc.

Text Books:

1. Anany Levitin: "Introduction to The Design and Analysis of Algorithms", (Chapters 1-5,7,9,11), Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN: 10: 8173716129.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI, London, England, 3rd Edition, 2010, ISBN: 9780262033848.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: "Introduction to the Design and Analysis of Algorithms A Strategic Approach", McGraw-Hill Higher Education, USA, International Edition, 2005, ISBN-13: 978-0071243469.

E-Resources:

1. http://www.pearsonhighered.com
2. http://www.ebooks.com
3. http://www.worldcat.org
4. http://www.citc.ui.ac.ir/zemoni/cls.pdf

Microprocessors (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI44	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basics of 8086 microprocessor family and its architecture.
- Become aware of constructing the machine codes for 8086 instructions.
- Learn 8086 instruction sets as needed to solve programming problems.
- Acquire knowledge about signals, timing and system connections for a simple 8086-based microcomputer.
- Understand how the 8086 responds to interrupts, how interrupts-service procedures are written and operation of peripheral devices.

Syllabus

Module - I

Computers, Microcomputers and Microprocessors- An Introduction: Types of Computers, How Computers and Microcomputers are used, an example, Overview of Microcomputer Structure and Operation, Microprocessor Evolution and Types, The 8086 Microprocessor Family-Overview, 8086 Internal Architecture, Introduction to Programming the 8086. **08 Hours**

Module - II

8086 Family Assembler Language Programming-I: Constructing the machine codes for 8086 Instructions (MOV Instructions only), Assembly Language Program Development Tools, Data Transfer Instructions, Arithmetic Instructions. **08 Hours**

Module - III

8086 Family Assembler Language Programming-II: Bit-manipulation Instructions, String Instructions, Program-Execution Transfer Instructions, Processor Control Instructions. **08 Hours**

Module - IV

8086 System Connection Timing, Interrupts and Interrupt Applications: A Basic 8086 Microcomputer System, 8086 Bus activities during a read machine cycle, 8086 Bus activities during a write machine cycle, A closer look at the 8086, 8086 Interrupts and Interrupt Responses, Hardware Interrupt Applications, 8259 A Priority Interrupt Controller, Overview and System Connections, Software Interrupt Applications.

08 Hours

Module - V

Digital Interfacing: Programmable Parallel Ports and Handshake Input/Output, Methods of Data Transfer, Implementing Handshake Data Transfer, 8255A Internal Block Diagram and System Connections, 8255A Operational Modes and Initialization, Constructing and Sending 8255A Control Words. **08 Hours**

Laboratory

1. a) Search a key element in a list of 'n' 16-bit numbers using the Binary search algorithm.
   b) Read the status of eight input bits from the Logic Controller Interface and display 'FF' if the parity of the input read is even; otherwise display '00'.
2. a) Write two ALP modules stored in two different files; one module is to read a character from the keyboard and the other one is to display a character. Use the above two modules to read a string of characters from the keyboard terminated by the carriage return and print the string on the display in the next line.
   b) Implement a BCD Up-Down Counter on the Logic Controller Interface.
3. a) Sort a given set of 'n' numbers in ascending order using the Bubble Sort algorithm.
   b) Read the status of two 8-bit inputs (X and Y) from the Logic Controller Interface and display X*Y.
4. a) Display the message "Welcome To Microprocessor Lab" at the centre of the screen, clear the screen before displaying the message.
   b) Display messages FIRE and HELP alternately with flickering effects on a 7-segment display Interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).
5. a) Reverse a given string and check whether it is a palindrome or not.
   b) Assume any suitable message of 12 characters length and display it in the rolling fashion on a 7-segment display interface for a suitable period of time Ensure a flashing rate that makes it easy to read both the messages. (Examiner does not specify these delay values nor is it necessary for the student to compute these values).
6. a) Read your name from the keyboard and display it at a specified location on the screen after the message "What is your name?", you must clear the entire screen before display.

b) Drive a Stepper Motor interface to rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

7. a) Compute ${}^{n}C_{r}$ using recursive procedure. Assume that 'n' and 'r' are non negative integers.

   b) Drive a stepper motor interface to rotate by N steps clockwise direction and N steps counter-clockwise direction. (N is specified by the examiner) Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

8. a) Generate the first 'n' Fibonacci numbers.

   b) Generate a Sine wave form using the DAC interface. (The output of the DAC is to be displayed on the CRO).

9. a) Read the current time from the system and display it in the standard format on the screen.

   b) Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).

10. a) Write a program to create a file (input file) and to delete an existing file.

    b) Generate the Fully Rectified Sine waveform using DAC interface (The output of the DAC is to be displayed on the CRO).

Course Outcomes:

On completion of this course, the students are able to :

- Discuss the microprocessor evolution, types and 8086 internal architecture.
- Construct the machine codes for 8086 instructions with the help of instruction templates.
- Describe functions of different types of 8086 assembly language instructions and use assembly language program to solve simple problems.
- Explain 8086 interrupt types and their applications.
- Describe the operations of peripheral devices.

Text Book:

1. Douglas V. Hall: "Microprocessors and Interfacing", (Chapter 2,3,5-9), TMH, New Delhi, Revised 2nd Edition, 2006, ISBN-13: 978-0070257429.

Reference Book:

1. Barry B. Brey: "The Intel Microprocessors", Pearson Education, New Delhi, 8th Edition, 2009, ISBN-13: 978-8131726228.

E-Resources:

1. http://www.nptel.ac.in/downloads/106108100/
2. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22
3. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22#tbm=bksandq=Bary+b+brey
4. http://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/Microprocessors%20and%20Microcontrollers/pdf/Teacher_Slides/mod1/M1L3.pdf
5. http://www.nptel.ac.in/downloads/106108100/
6. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22
7. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22#tbm=bksandq=Bary+b+brey
8. http://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/Microprocessors%20and%20Microcontrollers/pdf/Teacher_Slides/mod1/M1L3.pdf

UNIx and Shell Programming (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI451	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to **:**

- Understand the history, origin, features and architecture of UNIX Operating System.
- Learn basic commands to interact with UNIX System and VI editor.
- Understand UNIX file system.
- Learn shell scripting and Perl scripting.

Syllabus

Module - I

UNIX architecture: The operating system, the UNIX operating system, Linux and GNU. The UNIX architecture, features of UNIX, POSIX and Single UNIX specification, Internal and External commands, Command structure, man browsing and manual pages on-line.

The file system: The parent – child relationship, the HOME variable, pwd, cd, mkdir, absolute pathname, relative pathname.

vi editor: vi basics, input mod, saving text and quitting, searching for a pattern(| and ?), substitution- search and replace(:s). **08 Hours**

Module - II

Basic file attributes: ls: listing directory contents, the UNIX file system, ls –l, -d option, file ownership, file permissions, chmod, directory permissions, changing file ownership.

More file attributes: File systems and inodes, hard links, symbolic links and ln, the directory, umask, modification and access times, find.

The Process: Process basics, ps: process status, system processes(-e or –a), mechanism of process creation, process states and zombies, running jobs in background, nice: job execution, job control, at and batch, cron, time. **08 Hours**

Module - III

Simple filters: pr, head, tail, cut, paste, sort, uniq, tr.

Filters using regular expressions – grep and sed: grep, Basic Regular Expressions (BRE), Extended Regular Expressions (ERE) and egrep, sed: the stream editor, line

addressing using multiple instructions (-E and -F) context addressing, writing selected lines to a file (w), text editing, substitution (s), basic regular expression revisited.

08 Hours

Module - IV

The shell: The shell's interpretive cycle, shell offerings, pattern matching, escaping and quoting, redirection, pipes, tee, command substitution, shell variables.

Essential shell programming: Shell scripts, read using command line arguments, exit and exit status of command, the logical operators && and ||, the if conditional, using test and {} to evaluate expression. The case conditional, expr, $0, while, for, debugging shell scripts with set –x. **08 Hours**

Module - V

Perl – The master manipulator: Perl preliminaries, the chop function, variables and operators, the string handling functions, specifying file names in command line, $_, current line number ($.) and the range (..), list and arrays, foreach, split, join, dec2bin. pl, grep, associative arrays, regular expressions and substitutions, file handling, file tests, subroutines.

Networking Tools: TCP/IP basics, the applications. **08 Hours**

Laboratory

1. Execute the following list of basic commands in UNix:
   (i) pwd (ii) mkdir (iii) cd (iv) who (v) echo
   (vi) cat (vii) rm (viii) mv (ix) wc (x) cp
2. Execute the basic file attributes with all possible options:
   (i) ls (ii) chmod
3. Execute basic commands using vi editor:
   a. input mode commands
   b. saving text and quitting
   c. navigation
   d. editing text
   e. searching pattern
4. Execute the following filters using regular expressions with all possible options:
   (i) grep (ii) sed
5. Write a shell script to display current date and calendar.
6. Write a shell script to search for a pattern using grep in an employee database using interactive way.

7. Write a shell script to search for a pattern in an employee database using command line argument.
8. Write a shell script using test, $0 and $# in an if-elif-if construct.
9. Write a shell script using case, to offer 5 item menu:
    a. List of files
    b. Process of user
    c. Today's date
    d. Users of system
    e. Quit to UNIX
10. Write a shell script for matching pattern in student list using for loop.

Course Outcomes:

On completion of this course, the students are able to :

- Describe history, origin, feature and architecture of UNIX operating system.
- Interact with UNIX system easily.
- Construct and edit files, search for any patterns using regular expressions.
- Solve complex jobs using tools and utilities available in UNIX.
- Design and develop various tasks by using Shell and Perl scripting.

Text Book:

1. Sumitabha Das: "UNIX – Concepts and Applications", (Chapters 1,2,4,6- 9,11-14,17,19), Tata McGraw Hill, Noida, 4th Edition, 15th Reprint, 2011, ISBN-13: 978-0-07-063546-3.

Reference Books:

1. Behrouz A. Forouzan and Richard F. Gilberg: "UNIX and Shell programming", Cengage Learning, India, 1st Edition, 2005, ISBN: 81-35-0325-9.
2. M G Venkatesh Murthy: "UNIX and Shell programming", Pearson Education, Delhi, 1st Edition, 2005, ISBN: 81-7758-745-5.

E-Resources:

1. http://www.mhhe.com/das/uca
2. http://www.tutorialspoint.com/unix/unix_tutorials.pdf.
3. http://www.perldoc.perl.org/

Object Oriented Programming with C++ (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI452	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives

This course will enable students to :

- Introduced to the basic principles of object oriented programming.
- Understand the features of C++.
- Learn the concepts of operator overloading and friend functions.
- Acquire the knowledge of inheritance to build C++ programming.
- Understand different techniques like exception handling, virtual functions etc.

Syllabus

Module - I

Introduction: Overview of C++, What is object oriented programming, Some C++ fundamentals, Overview of pointers and user defined types, Functions: argument passing, in line functions, function overloading.

Class and objects: Class Specification, Class Objects, Scope resolution operator, Access members, Defining member functions. **08 Hours**

Module - II

Classes and Objects-1: Constructors, Destructors, Parameterized constructors, Static data members and functions, Friend functions, Passing objects as arguments, Returning objects, Arrays of objects, Dynamic objects, Pointers to objects, Copy constructors, Generic functions and classes with examples. **08 Hours**

Module - III

Classes and Objects-2: Operator overloading using member functions such as +, - , pre-increment, post-increment, Operator overloading using friend functions such as +, - , pre-increment, post-increment [ ], <<, >>.

Inheritance: Base Class access control, Inheritance and protected members, Protected base class inheritance, Inheriting multiple base classes. **08 Hours**

Module - IV

Inheritance: Constructors, Destructors and Inheritance, Passing parameters to base class constructors, Granting access, Virtual base classes.

Virtual functions, Polymorphism: Virtual function, Calling a Virtual function through

a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, Pure virtual functions, Early v/s late binding. **08 Hours**

Module - V

I/O System Basics, File I/O: C++ stream classes, Formatted I/O, I/O manipulators, fstream and the File classes, File operations

Exception Handling: Exception handling fundamentals, Exception handling option, Overview of STL. **08 Hours**

Laboratory

List of Experiments:

1. Design, develop, and execute a program in C++ based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_ Name (a string of characters), Basic_Salary (an integer) ,All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic; Income Tax (IT) = 30% of the gross salary (= basic_Salary _ All_Allowance); Net_Salary = Basic_Salary + All_ Allowances – IT).

2. a. Define a student class with USN, Name and Marks in 3 tests of a subject declare an array of 10 student objects using appropriate functions, Find the average of 2 better marks for each student, print the USN, Name and the average marks of all the students.

   b. Simulation of the above program using virtual lab.

3. Write a C++ program to create a class called complex and implement the following overloading function. ADD that return a complex number.

   a. ADD(a, S2)—where a is an integer(real part) and S2 ia a complex number.

   b. ADD(S1, S2)—where S1 and S2 are complex numbers.

4. Design, develop, and execute a program in C++ to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

   i. STRING s1 = “NCET”

   ii. STRING s2 = “BENGALURU”

   iii. STIRNG s3 = s1 + s2; (Use copy constructor)

5. Design, develop, and execute a program in C++ to create a class called OCTAL, which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.

   OCTAL h = x ; where x is an integer

   int y = h + k ; where h is an OCTAL object and k is an integer.

   Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

6. a. Write a C++ program to create a template function for quick sort and demonstrate sorting of integers and doubles.

   b. Simulation of the above program by using virtual lab.

7. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance create the classes UGSTUDENT and PGSTUDENT having fields as Semester, fees and stipend. Enter the data at least for 5 age for all UG and PG students separately.

8. Write a C++ program to create a class called MATRIX using a two dimensional array of integers. Implement the following operations by overloading the operator == which checks the compatibility of two matrices m1 and m2 to be added and subtracted. Perform the addition and subtraction by overloading the operators + and – respectively. Display the results by overloading operator <<.

```
if(m1==m2)
 {
    m3=m1+m2;
    m4=m1-m2;
 }
else
    Display error.
```

Course Outcomes:

On completion of this course, the students are able to :

- Describe the basic principles of object oriented programming language using C++.
- Illustrate class and objects, inheritance and polymorphism features, abstract classes.
- Construct C++ programs using operator overloading.

- Design and develop object oriented programming using inheritance.
- Interpret the basic concepts of I/O systems, file I/O and exception handling.

Text Book:

1. Herbert Schildt: "The Complete Reference C++", (Chapters 1-5), Tata McGraw Hill, US, 5th Edition, 2014, ISBN-10: 00716348 00, ISBN – 13: 9780071634809.

Reference Books:

1. Stanley B. Lippmann, Josee Lajore: "C++ Primer", Pearson Education, US, 5th Edition, 2012, ISBN 10: 8131775275, ISBN - 13: 9788131775271.
2. Paul J Deitel, Harvey M Deitel: "C++ for Programmers", Pearson Education, US, 7th Edition, 2010, ISBN 10: 0132165414, ISBN 13: 9780132165419.
3. K R Venugopal, Rajkumar Buyya, T Ravi Shankar: "Mastering C++", Tata McGraw Hill, New Delhi, 4th Edition, 1999, ISBN-10/AS1: 0074634542, ISBN-13: 9780074634547.

E-Resources:

1. https://docs.google.com/file/d/OByYLrayxuOPHyvJHCEFtcFFCNVE/edit.
2. http://books.google.co.in/books?
3. www.deitel@deitel.com
4. http://freedomputurbooks.com/masterin'sccp-by-venugopal.html
5. www.cplusplus.com
6. www.doc.ic.ac.uk
7. www.pearsonhighered.com/savitech

Introduction to Programming using Python (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI453	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course enable students to :

- To get a clear understanding of Object Oriented Programming.
- To learn Python with a focus on regular expressions, exception handling, file handling, creating modules, interacting with database.

Syllabus

Module - I

Introduction and overview: What is Python, Origin, Comparison, Comments, Operators, Variables and Assignment, Numbers, Strings, Lists and Tuples, Dictionaries, if Statement, while Loop, for Loop and the range() built-in Function, Files and the open() Built-in Function, Errors and Exceptions, Functions, Classes, Modules.

Syntax and Style: Statements and Syntax, Variable Assignment, Identifiers, Basic Style Guidelines, Memory Management, Python Application Examples. **08 Hours**

Module - II

Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types. Numbers and Strings Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions.

Sequences: Strings, Lists, and Tuples, Sequences, Strings, Strings and Operators, String-only Operators, Built-in Functions, String Built-in Methods, Special Features of Strings. **08 Hours**

Module - III

Lists Operators, Built-in Functions, List Type Built-in Methods, Special Features of Lists, Tuples, Tuple Operators and Built-in Functions, Special Features of Tuples. Conditionals and Loops if statement, else if Statement, while Statement, for Statement, break Statement, continue Statement, pass Statement, else Statement.

08 Hours

Module - IV

Files and Input/output File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules.

Exception handling: The dir Function, Errors, Runtime Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions, raise, assert. **08 Hours**

Module - V

Regular Expressions: Introduction/Motivation, Special Symbols and Characters for REs, REs and Python.

Programming Exercise: Check for data error in CSV files: Numeric Check, Alphanumeric Check, Email Check, Date Check.

Database Interactions: Database Connection, creating database tables, insert data into table, reading, updating data. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Describe basic of object oriented principles in Python.
- Explain type of operators and Built-in functions.
- Identify built-in libraries and conditional statements like loops.
- Interpret file handling and exception handling mechanisms.
- Use regular expressions and apply the same in solving specific problems.

Text Book:

1. Mark Lutz: "Learning Python", O'REILLY, 5th Edition, 2013, ISBN: 978-4493-5573-9.

Reference Books:

1. Barry, Paul: "Head First Python", O'REILLY, 2nd Edition, 2010, ISBN: 978-1-4493-8267-4.
2. David M. Beazley: "Python Essential Reference", Developer's Library, 4th Edition, 2010, ISBN: 0672329786.

E-Resources:

1. http://www.tutorialspoint.com/python/
2. https://www.codementor.io/learn-python-online
3. https://www.youtube.com/playlist?list=PL9FAE4422FA13FDE4

Introduction to Cyber Security and Cyber Laws

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST461	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

- Understand the area of cyber crime and forensics.
- Understand the motive and causes for cyber crime, detection and handling.
- Study the areas affected by cyber crime and investigation.
- Understand the tools used in cyber forensic.
- Know Legal Perspectives in cyber security.

Syllabus

Module - I

Introduction to Cyber crime: Cyber crime: Definition and Origins of the Word, Cyber crime and Information Security, Who are Cyber criminals?, Classifications of Cyber crimes, Cyber crime: The Legal Perspectives, Cyber crimes: An Indian Perspective, Cyber crime and the Indian ITA 2000, A Global Perspective on Cyber crimes, Cyber crime Era: Survival Mantra for the Netizens.

Cyber Offenses: How Criminals Plan Them: How Criminals Plan the Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cyber crimes, Bot nets: The Fuel for Cyber crime, Attack Vector, Cloud Computing. **08 Hours**

Module - II

Cyber crime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/CellPhones, Mobile Devices: Security Implications for organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops. **08 Hours**

Module - III

Tools and Methods Used in Cyber crime: Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spy wares, Virus and Worms, Trojan Horses and Back doors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks.

Phishing and Identity Theft: Introduction, Phishing, Identity Theft (ID Theft).

08 Hours

Module - IV

Understanding Computer Forensics: Introduction, Historical Background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber

forensics and Digital Evidence, Forensics Analysis of E-Mail, Digital Forensics Life Cycle, Chain of Custody Concept, Network Forensics, Approaching a Computer Forensics Investigation, Setting up a Computer Forensics Laboratory: Understanding the Requirements, Computer Forensics and Steganography, Relevance of the OSI 7 Layer Model to Computer Forensics, Forensics and Social Networking Sites: The Security/Privacy Threats, Computer Forensics from Compliance Perspective, Challenges in Computer Forensics, Special Tools and Techniques, Forensics Auditing, Anti forensics. **08 Hours**

Module - V

Cyber crimes and Cyber security: The Legal Perspectives, Introduction, Cyber crime and the Legal Landscape around the World, Why Do We Need Cyber laws: The Indian Context, The Indian IT Act, Challenges to Indian Law and Cyber crime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cyber crime and Punishment, Cyber law, Technology and Students: Indian Scenario.

08 Hours

Course Outcomes:

On completion of this course, the students are able to :

- Acquire knowledge about the cyber security cyber crime and cyber offenses.
- Explain cyber crime on various mobile and wireless devices.
- Use of tools and methods in cyber crime and security.
- Interpret computer forensics.
- Understand legal issues in cyber crime.

Text Book:

1. Sunit Belapure and Nina Godbole: "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", (Chapter 1-7), Wiley India Pvt. Ltd., New Delhi, India, 2011, ISBN: 8126521791.

Reference Books:

1. Dr. Surya Prakash Tripathi, Ritendra Goyal, Praveen Kumar Shukla, KLSI: "Introduction to information security and cyber laws", Dreamtech Press, New Delhi, India, 2014, ISBN 13: 789351194736.

E-Resources:

1. http://www.civilserviceindia.com/subject/General-Studies/notes/basics-of-cyber security.html
2. http://uttaminstitute.ac.in/CYBER.pdf
3. http://www.vssut.ac.in/lecture_notes/lecture1423183198.pdf
4. http://www.tutorialspoint.com/information_security_cyber_law/introduction.html

Linear Integrated Circuits

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST462	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

- Interpret the basic building blocks of linear integrated circuits.
- Learn the linear and non-linear applications of operational amplifiers.
- Acquire the knowledge about the theory and applications of analog multipliers and PLL.
- Understand the theory of ADC and DAC.
- Learn the concepts of waveform generation and introduce some special function ICs.

Syllabus

Module - I

IC Fabrication and Circuit Configuration for Linear IC: Advantages of ICs over discrete components, Manufacturing process of monolithic ICs , Construction of monolithic bipolar transistor, Monolithic diodes, Integrated resistors, Monolithic capacitors, Inductors, Current mirror and current sources, Current sources as active loads, Voltage sources, Voltage references, BJT Differential amplifier with active loads, General operational amplifier stages, and internal circuit diagrams IC 741, DC and AC performance characteristics, slew rate, Open and closed loop configurations.

08 Hours

Module - II

Applications of Operational Amplifiers: Sign changer, Scale changer, Phase shift circuits, Voltage follower, V-to-I and I-to-V converters, Adder, Subtractor, Instrumentation amplifier, Integrator, Differentiator, Logarithmic amplifier, Antilogarithmic amplifier, Comparators, Schmitt trigger, Precision rectifier, Peak detector, Clipper and Clamper, Low-pass, High-pass and band-pass butter worth filters.

08 Hours

Module - III

Analog Multiplier and PLL: Analog multiplier using emitter coupled transistor pair, Gilbert multiplier cell, Variable trans conductance technique, Analog multiplier ICs and their applications, Operation of the basic PLL, Closed loop analysis, Voltage

controlled oscillator, Monolithic PLL IC 565, Application of PLL for AM detection, FM detection, FSK modulation and demodulation and frequency synthesizing. **08 Hours**

Module - IV

Analog to Digital and Digital to Analog Converters: Analog and digital data conversions, D/A converter, specifications, weighted resistor type, R-2R ladder type, Voltage mode and current mode R – 2R ladder types, switches for D/A converters, high speed sample-and hold circuits, A/D converters, specifications, Flash type, Successive approximation type, Single slope type, Dual slope type, A/D converter using voltage-to-Time conversion, Over-sampling A/D converters. **08 Hours**

Module - V

Waveform Generators and Special Function ICs: Sine-wave generators, Multi vibrators and triangular wave generator, Saw-tooth wave generator, ICL8038 function generator, Timer IC 555, IC voltage regulators – Three terminal fixed and adjustable voltage regulators – IC 723 general purpose regulator – Monolithic switching regulator, Switched capacitor filter IC MF10, Frequency to voltage and voltage to frequency converters, Audio power amplifier, Video amplifier, Isolation amplifier, Opto-couplers and fibre optic IC. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Implement linear and non-linear applications of OP-AMPs.
- Design applications using analog multiplier and PLL.
- Illustrate the working of ADC and DAC using OP-AMPs.
- Generate waveforms using OP-AMPs circuits.
- Analyze special function ICs.

Text Books:

1. Sergio Franco: “Design with operational amplifiers and analog integrated circuits”, (Chapters 1-4,6,7,9-11), Tata McGraw-Hill, 4th Edition, 2015, ISBN: 978-0078028168.

2. D. Roy Choudhry, Shail Jain: “Linear Integrated Circuits”, (Chapters 4-10), New Age International Pvt. Ltd., 4th Edition, 2010, ISBN 10: 8122430988 ISBN 13: 9788122430981.

Reference Books:

1. Gray and Meyer: " Analysis and Design of Analog Integrated Circuits", Wiley International, 5th Edition, 2014, ISBN 10 : 8126521481, ISBN 13: 978-81-265-2148-7.
2. Ramakant A. Gayakwad: "OP-AMP and Linear ICs", Prentice Hall / Pearson Education, 4th Edition, 2002, ISBN: 8120320581, 9788120320581.

E-Resources:

1. https://books.google.co.in/books?isbn=0070530440
2. https://books.google.co.in/books?isbn=8122414702

Control Systems

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST463	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

- Learn the concepts of block diagrams and transfer functions.
- Understand the characteristics of closed loop control systems including steady state and transient response.
- Acquire the knowledge about basic performance criteria for 1st and 2nd order systems.
- Understand the basic control system design methods including root-locus diagrams and frequency response methods.

Syllabus

Module - I

Modelling of Systems: Introduction to Control Systems, Types of Control Systems, Effect of Feedback Systems, Differential equation of Physical Systems -Mechanical systems, Friction, Translational systems (Mechanical accelerometer, systems excluded), Rotational systems, Gear trains, Electrical systems, Analogous systems.

08 Hours

Module - II

Block diagrams and signal flow graphs: Transfer functions, Block diagram algebra, Signal Flow graphs (State variable formulation excluded). **08 Hours**

Module - III

Time Response of feed back control systems: Standard test signals, Unit step response of First and second order systems, Time response specifications, Time response specifications of second order systems, Steady – state errors and error constants. Introduction to PID Controllers (excluding design). **08 Hours**

Module - IV

Stability analysis: Concepts of stability, Necessary conditions for Stability, Routh-stability criterion.

Root-Locus Techniques: Introduction, The root locus concepts. **08 Hours**

Module - V

Frequency domain analysis: Correlation between time and frequency response, Bode plots, Experimental determination of transfer functions, Assessment of relative stability using Bode Plots. Introduction to lead, Lag and lead-lag compensating networks (excluding design). **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Describe the engineering fundamentals about block diagram and transfer functions.
- Identify the characteristics of closed loop control systems.
- Examine time response of 1st and 2nd order feedback control systems.
- Estimate stability analysis using R-H criteria and root locus concepts.
- Construct correlation between time and frequency responses

Text Book:

1. J. Nagrath and M. Gopal: "Control Systems Engineering", (Chapter 2,5-8, 10), New Age International (P) Limited Publishers, Delhi, 5th Edition, 2008, ISBN: 978-184-8290037.

Reference Books:

1. K. Ogata: "Modern Control Engineering", Pearson Education Asia/ PHI, Delhi, 5th Edition, 2002, ISBN 10: 0-13-615673-8, ISBN 13: 978-0-13-615673-4.
2. Benjamin C. Kuo, John: "Automatic Control Systems", Wiley India Pvt. Ltd., 8th Edition, 2008, ISBN-10: 0471134767, ISBN-13: 978-047113476.
3. Joseph J Distefano III et al.,: "Feedback and Control System", P Schaum's Outlines, TMH, McGraw-Hill, New York, 2nd Edition, 2007, ISBN: 0-07-017052-5.

E-Resources:

1. http://research.iaun.ac.ir/pd/mahmoodian/pdfs/UploadFile_9809.pdf
2. http://www2.nuu.edu.tw/emo/e-teaching%20materials/Automatic%20Control.pdf
3. https://murdercube.com/files/Miscellaneous/Feedback%20and%20Control%20System.PDF

Design and Analysis of Algorithms Laboratory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSL47	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course enable students to :

- Understand and use asymptotic notations to analyze the performance of algorithms.
- Learn different sorting and searching techniques.
- Get exposed to various algorithm design techniques.

Design, develop and implement the specified algorithms for the following problems using C/C++ Language in LINUX / Windows environment.

1. a. Write a C/C++ program to sort the elements by using quick sort method.
   b. Simulation of the above program by using virtual lab.
2. a. Write a C/C++ program to sort the elements by using merge sort method.
   b. Simulation of the above program by using virtual lab.
3. Obtain the Topological ordering of vertices in a given digraph.
4. a. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
   b. Perform Dijkstra single source shortest path on a graph using virtual lab.
5. Implement 0/1 Knapsack problem using Dynamic Programming.
6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
8. a. Compute the transitive closure of a given directed graph using Warshall's algorithm.
   b. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
9. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
   b. Check whether a given graph is connected or not using DFS method.
10. Implement N Queen's problem using Back Tracking.

Course Outcomes:

On completion of this course, the students are able to :

- Solve problems by applying appropriate algorithms.
- Analyze the efficiency of various algorithms.
- Apply techniques of stacks and queues to solve problems.
- Develop a program that can be solved in many ways using different techniques.
- Identify and evaluate complex problems using principles of maths and engineering science.

Text Books:

1. Anany Levitin: "Introduction to the Design and Analysis of Algorithms", Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", Universities Press, Hyderabad, 2nd Edition, 2007, ISBN-10: 8173716129.

E-Resources:

1. http://cs.gmu.edu/~pwiegand/cs483-Spring06/lecturenotes/cs483-l1pf.pdf
2. http://www.cs.cornell.edu/~kozen/papers/daa.pdf

Cloud Computing Foundations (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI48	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course enable students to :

- Explain the phases of transition from classic data center to virtual data center and then to the Cloud
- Explain the key characteristics, services, and deployment models of Cloud
- Describe the Cloud infrastructure components and service management processes
- Describe the Cloud security concerns and solutions
- List the key considerations for migration to the Cloud

Syllabus

Module - I

Journey to the Cloud: Drivers for cloud computing, cloud definition and characteristics, building cloud infrastructure – a phased approach from Classic data center to virtual data center to Cloud, virtualization and its benefits. **02 Hours**

Module - II

Cloud Computing Primer: Cloud computing characteristics, cloud deployment models private, public, hybrid and community cloud, cloud services – SaaS, PaaS, and IaaS, cloud economics and challenges. **03 Hours**

Module - III

Cloud infrastructure and Management: Cloud infrastructure framework and components, infrastructure management and service creation tools, cloud service management processes asset and configuration management, service catalog management, financial management, capacity, performance and availability management, incident, problem and compliance management. **04 Hours**

Module - IV

Cloud Security: Basic information security concepts, cloud security concerns and threats, security mechanisms in cloud at compute, storage, and network layer, Governance, Risk and compliance in Cloud. **02 Hours**

Module - V

Cloud Migration Considerations: Considerations for choosing right application and cloud model, service provider specific considerations, cloud adoption phases, Financial and technical feasibility assessment, migration and optimization considerations. **03 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Explain the phases of transition from classic data center to virtual data center and then to the Cloud.
- Explain the key characteristics, services, and deployment models of Cloud.
- Describe the Cloud infrastructure components and service management processes.
- Describe the Cloud security concerns and solutions.
- List the key considerations for migration to the Cloud.

Reference Books:

1. Thomas Erl: "Cloud Computing", Pearson Education, 1st Edition, 2014, ISBN-13: 978-9332535923.
2. Judith Hurwitz, Marcia Kaufman, Fern Halper: "Cloud Computing for dummies", Wiley, 1st Edition, 2009, ISBN-13: 978-0470484708.

E-Resources:

1. http://www.buyya.com/MasteringClouds/ToC-Preface-TMH.pdf

Soft Skills Development

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSH49	0:2:0:0	1	CIE:50 SEE:50	2 Hours	HSS

Course Objectives:

This course will enable students to :

- Improve the communication skills.
- Develop the art of presentation and writing effectively.
- Enhance the technical knowledge.

To improve the communication and presentation skills, every student has to give a seminar on technical topics assigned by the supervisors. Each course coordinator/ faculty members will be assigned with few students to guide and monitor the presentation. The presentation shall be for 15 minutes. A brief report on the seminar has to be submitted by the student to the concerned department after completion of the seminar. The report shall be signed by the supervisor and the Head of the concerned department.

The objective of the seminar is to introduce students to the major constituent of technology that is concerned with critically reading, understanding, summarizing, explaining and presenting existing technical topics. Students have to refer one or more topics that are assigned to them by their supervisors. The idea behind the seminar system is to familiarize student more extensively with the methodology of their chosen subject, allow them to develop presentation skills, and also interact with example of practical problems.

Course Outcomes:

On completion of this course, the students are able to :

- Get rid of stage fear and answer questions from audience.
- Communicate confidently and fluently.
- Comprehend and prepare reports effectively.

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, Tata McGraw Hill, 2007.
2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India, 2009.

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, Pearson Education, 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, 8th edition, Wiley India, 2009.
   (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: Introduction to Operating Systems: Concepts and Practice, 2nd Edition, PHI, 2008.
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, 6th Edition, Mc-GrawHill, 2010.
2. C.J. Date, A. Kannan, S. Swamynatham: An 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 **7 Hours**

Introduction: Data Communications, Networks, The Internet, Protocols & Standards, Layered Tasks,
The OSI model, Layers in OSI model, TCP/IP Protocol suite, Addressing

UNIT- 2 **7 Hours**

Physical Layer-1: Analog & Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital-digital conversion (Only Line coding: Polar, Bipolar and Manchester coding), Analog-to-digital conversion (only PCM), Transmission Modes, Digital-to-analog conversion

UNIT- 3 **6 Hours**

Physical Layer-2 and Switching: Multiplexing, Spread Spectrum, Introduction to switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks

UNIT- 4 **6 Hours**

Data Link Layer-1: Error Detection & Correction: Introduction, Block coding, Linear block codes, Cyclic codes, Checksum.

PART - B

UNIT- 5 **6 Hours**

Data Link Layer-2: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy channels, HDLC, PPP (Framing, Transition phases only)

UNIT- 6 **7 Hours**

Multiple Access & Ethernet: Random access, Controlled Access, Channelization, Ethernet: IEEE standards, Standard Ethernet, Changes in the standard, Fast Ethernet, Gigabit Ethernet

UNIT - 7 **6 Hours**

Wireless LANs and Cellular Networks: Introduction, IEEE 802.11, Bluetooth, Connecting devices, Cellular Telephony

UNIT - 8: **7 Hours**

Network Layer: Introduction, Logical addressing, IPv4 addresses, IPv6 addresses, Internetworking basics,
IPv4, IPv6, Comparison of IPv4 and IPv6 Headers.

Text Books:

1. Behrouz A. Forouzan,: Data Communication and Networking, 4th Edition Tata McGraw-Hill, 2006.
   (Chapters 1.1 to 1.4, 2.1 to 2.5, 3.1 To 3.6, 4.1 to 4.3, 5.1, 6.1, 6.2, 8.1 to 8.3, 10.1 to 10.5, 11.1 to 11.7, 12.1 to 12.3, 13.1 to 13.5, 14.1, 14.2, 15.1, 16.1, 19.1, 19.2, 20.1 to 20.3)

Reference Books:

1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks - Fundamental Concepts and Key architectures, 2nd Edition Tata McGraw-Hill, 2004.
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. 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 Learning, 2009.
2. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation, Principles and Practice, Elsevier, 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 (f*id*: 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 ($a^n b^n$, n>= 0).

6. Program to recognize the grammar ($a^n b$, 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 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.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: The New Frontiers, Pearson Education, 2001.

COMPILER DESIGN

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

PART – A

UNIT – 1 **8 Hours**

Introduction, Lexical analysis: Language processors; The structure of a Compiler; The evolution pf 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 Louden: 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 Neworks: 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 to18.3, 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	**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, 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	**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: 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-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, 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:

1. Alan V. Oppenheim, Alan S. Willsky and S. Hamid Nawab: Signals and Systems, 2nd Edition, PHI, 1997, Indian reprint 2009.
2. Ganesh Rao D and Satish Tunga: Signals and Systems - A Simplified Approach, Sanguine Technical Publishers, 2003-04.

DATA COMPRESSION

Subject Code: 10CS663 **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	**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.

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	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**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.

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 SYSTEM 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, 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.

VII SEMESTER

OBJECT-ORIENTED MODELING AND DESIGN

Subject Code: 10CS71	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **7 Hours**

Introduction, Modeling Concepts, class Modeling: What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history

Modeling as Design Technique: Modeling; abstraction; The three models.

Class Modeling: Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips.

UNIT – 2 **6 Hours**

Advanced Class Modeling, State Modeling: Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips.

State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips.

UNIT – 3 **6 Hours**

Advanced State Modeling, Interaction Modeling: Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips.

Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models.

UNIT – 4 **7 Hours**

Process Overview, System Conception, Domain Analysis: Process Overview: Development stages; Development life cycle.

System Conception: Devising a system concept; Elaborating a concept; Preparing a problem statement.

Domain Analysis: Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis.

PART – B

UNIT – 5 **7 Hours**

Application Analysis, System Design: Application Analysis: Application interaction model; Application class model; Application state model; Adding operations.

Overview of system design; Estimating performance; Making a reuse plan; Breaking a system in to sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example.

UNIT – 6 **7 Hours**

Class Design, Implementation Modeling, Legacy Systems: Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example.

Implementation Modeling: Overview of implementation; Fine-tuning classes; Fine-tuning generalizations; Realizing associations; Testing.

Legacy Systems: Reverse engineering; Building the class models; Building the interaction model; Building the state model; Reverse engineering tips; Wrapping; Maintenance.

UNIT – 7 **6 Hours**

Design Patterns – 1: What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description
Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber.

UNIT – 8 **6 Hours**

Design Patterns – 2, Idioms: Management Patterns: Command processor; View handler.
Idioms: Introduction; what can idioms provide? Idioms and style; Where to find idioms; Counted Pointer example

Text Books:

1. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education, 2005. (Chapters 1 to 17, 23)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007. (Chapters 1, 3.5, 3.6, 4)

Reference Books:

1. Grady Booch et al: Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.
2. Brahma Dathan, Sarnath Ramnath: Object-Oriented Analysis, Design, and Implementation, Universities Press, 2009.
3. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, Wiley- Dreamtech India, 2004.
4. Simon Bennett, Steve McRobb and Ray Farmer: Object-Oriented Systems Analysis and Design Using UML, 2nd Edition, Tata McGraw-Hill, 2002.

EMBEDDED COMPUTING SYSTEMS

Sub Code: 10CS72	**IA Marks :25**
Hrs/Week: 04	**Exam Hours :03**
Total Hrs: 52	**Exam Marks :100**

PART- A

UNIT – 1 **6 Hours**

Embedded Computing: Introduction, Complex Systems and Microprocessors, Embedded Systems Design Process, Formalism for System design

Design Example: Model Train Controller.

UNIT – 2 **7 Hours**

Instruction Sets, CPUs: Preliminaries, ARM Processor, Programming Input and Output, Supervisor mode, Exceptions, Traps, Coprocessors, Memory Systems Mechanisms, CPU Performance, CPU Power Consumption. Design Example: Data Compressor.

UNIT – 3 **6 Hours**

Bus-Based Computer Systems: CPU Bus, Memory Devices, I/O devices, Component Interfacing, Designing with Microprocessor, Development and Debugging, System-Level Performance Analysis

Design Example: Alarm Clock.

UNIT – 4 **7 Hours**

Program Design and Analysis: Components for embedded programs, Models of programs, Assembly, Linking and Loading, Basic Compilation Techniques, Program optimization, Program-Level performance analysis, Software performance optimization, Program-Level energy and power analysis, Analysis and optimization of program size, Program validation and testing. Design Example: Software modem.

PART- B

UNIT – 5 **6 Hours**

Real Time Operating System (RTOS) Based Design – 1: Basics of OS, Kernel, types of OSs, tasks, processes, Threads, Multitasking and Multiprocessing, Context switching, Scheduling Policies, Task Communication, Task Synchronization.

UNIT – 6 **6 Hours**

RTOS-Based Design - 2: Inter process Communication mechanisms, Evaluating OS performance, Choice of RTOS, Power Optimization. Design Example: Telephone Answering machine

UNIT – 7 **7 Hours**

Distributed Embedded Systems: Distributed Network Architectures, Networks for Embedded Systems: I2C Bus, CAN Bus, SHARC Link Ports, Ethernet, Myrinet, Internet, Network Based Design. Design Example: Elevator Controller.

UNIT – 8 **7 Hours**

Embedded Systems Development Environment: The Integrated Development Environment, Types of File generated on Cross Compilation, Dis-assembler /Decompiler, Simulators, Emulators, and Debugging, Target Hardware Debugging.

Text Books:

1. Wayne Wolf: Computers as Components, Principles of Embedded Computing Systems Design, 2nd Edition, Elsevier, 2008.
2. Shibu K V: Introduction to Embedded Systems, Tata McGraw Hill, 2009
   (Chapters 10, 13)

Reference Books:

1. James K. Peckol: Embedded Systems, A contemporary Design Tool, Wiley India, 2008
2. Tammy Neorgaard: Embedded Systems Architecture, Elsevier, 2005.

PROGRAMMING THE WEB

Subject Code: 10CS73	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

UNIT – 1 **6 Hours**

Fundamentals of Web, XHTML – 1: Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, The Web Programmers Toolbox.

XHTML: Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links.

UNIT – 2 **7 Hours**

XHTML – 2, CSS: XHTML (continued): Lists, Tables, Forms, Frames

CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The box model, Background images, The and tags, Conflict resolution.

UNIT – 3 **6 Hours**

Javascript: Overview of Javascript, Object orientation and Javascript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts, Examples.

UNIT – 4 **7 Hours**

Javascript and HTML Documents, Dynamic Documents with Javascript: The Javascript execution environment, The Document Object Model, Element access in Javascript, Events and event handling, Handling events from the Body elements, Button elements, Text box and Password elements, The DOM 2 event model, The navigator object, DOM tree traversal and modification.

Introduction to dynamic documents, Positioning elements, Moving elements, Element visibility, Changing colors and fonts, Dynamic content, Stacking elements, Locating the mouse cursor, Reacting to a mouse click, Slow movement of elements, Dragging and dropping elements.

PART - B

UNIT – 5 **6 Hours**

XML: Introduction, Syntax, Document structure, Document type definitions, Namespaces, XML schemas, Displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors, Web services.

UNIT – 6 **7 Hours**

Perl, CGI Programming: Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples.

The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module; A survey example; Cookies.

Database access with Perl and MySQL

UNIT – 7 **6 Hours**

PHP: Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control

statements, Arrays, Functions, Pattern matching, Form handling, Files, Cookies, Session tracking, Database access with PHP and MySQL.

UNIT – 8 **7 Hours**

Ruby, Rails: Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching.
Overview of Rails, Document requests, Processing forms, Rails applications with Databases, Layouts.

Text Books:

1. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson Education, 2008.
   (Listed topics only from Chapters 1 to 9, 11 to 15)

Reference Books:

1. M. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to Program, 4th Edition, Pearson Education, 2004.
2. Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2007.
3. Xue Bai et al: The web Warrior Guide to Web Programming, Cengage Learning, 2003.

ADVANCED COMPUTER ARCHITECTURES

Subject Code: 10CS74	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **6 Hours**

Fundamentals Of Computer Design: Introduction; Classes of computers; Defining computer architecture; Trends in Technology, power in Integrated Circuits and cost; Dependability; Measuring, reporting and summarizing Performance; Quantitative Principles of computer design.

UNIT – 2 **6 Hours**

Pipelining: Introduction; Pipeline hazards; Implementation of pipeline; What makes pipelining hard to implement?

UNIT – 3 **7 Hours**

Instruction –Level Parallelism – 1: ILP: Concepts and challenges; Basic Compiler Techniques for exposing ILP; Reducing Branch costs with

prediction; Overcoming Data hazards with Dynamic scheduling; Hardware-based speculation.

UNIT – 4 **7 Hours**

Instruction –Level Parallelism – 2: Exploiting ILP using multiple issue and static scheduling; Exploiting ILP using dynamic scheduling, multiple issue and speculation; Advanced Techniques for instruction delivery and Speculation; The Intel Pentium 4 as example.

PART - B

UNIT – 5 **7 Hours**

Multiprocessors and Thread –Level Parallelism: Introduction; Symmetric shared-memory architectures; Performance of symmetric shared–memory multiprocessors; Distributed shared memory and directory-based coherence; Basics of synchronization; Models of Memory Consistency

UNIT – 6 **6 Hours**

Review of Memory Hierarchy: Introduction; Cache performance; Cache Optimizations, Virtual memory

UNIT – 7 **6 Hours**

Memory Hierarchy design: Introduction; Advanced optimizations of Cache performance; Memory technology and optimizations; Protection: Virtual memory and virtual machines.

UNIT – 8 **7 Hours**

Hardware and Software for VLIW and EPIC: Introduction: Exploiting Instruction-Level Parallelism Statically; Detecting and Enhancing Loop-Level Parallelism; Scheduling and Structuring Code for Parallelism; Hardware Support for Exposing Parallelism: Predicated Instructions; Hardware Support for Compiler Speculation; The Intel IA-64 Architecture and Itanium Processor; Conclusions.

Text Books:

1. John L. Hennessey and David A. Patterson: Computer Architecture, A Quantitative Approach, 4th Edition, Elsevier, 2007.
   (Chapter. 1.1 to 1.9, 2.1 to 2.10, 4.1to 4.6, 5.1 to 5.4, Appendix A, Appendix C, Appendix G)

Reference Books:

1. Kai Hwang: Advanced Computer Architecture Parallelism, Scalability, Programability**,** 2nd Edition**,** Tata Mc Graw Hill, 2010.

2. David E. Culler, Jaswinder Pal Singh, Anoop Gupta: Parallel Computer Architecture, A Hardware / Software Approach, Morgan Kaufman, 1999.

ADVANCED DBMS

Subject Code: 10CS751	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **7 Hours**

Overview of Storage and Indexing, Disks and Files: Data on external storage; File organizations and indexing; Index data structures; Comparison of file organizations; Indexes and performance tuning

Memory hierarchy; RAID; Disk space management; Buffer manager; Files of records; Page formats and record formats

UNIT – 2 **7 Hours**

Tree Structured Indexing: Intuition for tree indexes; Indexed sequential access method; B+ trees, Search, Insert, Delete, Duplicates, B+ trees in practice

UNIT – 3 **6 Hours**

Hash-Based Indexing: Static hashing; Extendible hashing, Linear hashing, comparisons

UNIT – 4 **6 Hours**

Overview of Query Evaluation, External Sorting : The system catalog; Introduction to operator evaluation; Algorithms for relational operations; Introduction to query optimization; Alternative plans: A motivating example; what a typical optimizer does.

When does a DBMS sort data? A simple two-way merge sort; External merge sort

PART - B

UNIT – 5 **6 Hours**

Evaluating Relational Operators : The Selection operation; General selection conditions; The Projection operation; The Join operation; The Set operations; Aggregate operations; The impact of buffering

UNIT – 6 **7 Hours**

A Typical Relational Query Optimizer: Translating SQL queries in to Relational Algebra; Estimating the cost of a plan; Relational algebra equivalences; Enumeration of alternative plans; Nested sub-queries; other approaches to query optimization.

UNIT – 7 **7 Hours**

Physical Database Design and Tuning: Introduction; Guidelines for index selection, examples; Clustering and indexing; Indexes that enable index-only plans; Tools to assist in index selection; Overview of database tuning; Choices in tuning the conceptual schema; Choices in tuning queries and views; Impact of concurrency; DBMS benchmarking.

UNIT – 8 **6 Hours**

More Recent Applications: Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management

Text Books:

1. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2003.
   (Chapters 8, 9, 10, 11, 12, 13.1 to 13.3, 14, 15, 20)
2. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Pearson Education, 2007.
   (Chapter 30)

Reference Books:

1. Connolly and Begg: Database Systems, 4th Edition, Pearson Education, 2002.

DIGITAL SIGNAL PROCESSING

Subject Code: 10CS752	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **7 Hours**

The Discrete Fourier Transform: Its Properties and Applications : Frequency Domain Sampling: The Discrete Fourier Transform: Frequency Domain Sampling and Reconstruction of Discrete-Time Signals, The Discrete Fourier Transform (DFT), The DFT as a Linear Transformation, Relationship of the DFT to other Transforms. Properties of the DFT: Periodicity, Linearity and Symmetry Properties, Multiplication of Two DFT‟s and Circular Convolution, Additional DFT Properties; Linear Filtering

Methods Based on the DFT: Use of the DFT in Linear Filtering, Filtering of Long Data Sequences; Frequency Analysis of Signals using the DFT.

UNIT – 2 **7 Hours**

Efficient Computation of the DFT: Fast Fourier Transform Algorithms: Efficient Computation of the DFT: FFT Algorithms : Direct Computation of the DFT, Divide-and-Conquer Approach to Computation of the DFT, Radix-2 FFT Algorithms, Radix-4 FFT Algorithms, Split-Radix FFT Algorithms, Implementation of FFT Algorithms.

Applications of FFT Algorithms: Efficient computation of the DFT of Two Real Sequences, Efficient computation of the DFT of a 2N-Point Real Sequence, Use of the FFT Algorithm in Linear filtering and Correlation.

A Linear filtering approach to Computation of the DFT: The Goertzel Algorithm, The Chirp-Z Transform Algorithm.

Quantization Effects in the Computation of the DFT: Quantization Errors in the Direct Computation of the DFT, Quantization Errors in FFT Algorithms.

UNIT – 3 **6 Hours**

Implementation of Discrete-Time Systems – 1: Structures for the Realization of Discrete-Time Systems

Structures for FIR Systems: Direct-Form Structures, Cascade-Form Structures, Frequency-Sampling Structures, Lattice Structure.

Structures for IIR Systems: Direct-Form Structures, Signal Flow Graphs and Transposed Structures, Cascade-Form Structures, Parallel-Form Structures, Lattice and Lattice-Ladder Structures for IIR Systems.

UNIT – 4 **6 Hours**

Implementation of Discrete-Time Systems – 2: State-Space System Analysis and Structures: State-Space Descriptions of Systems Characterized by Difference Equations, Solution of the State-Space Equations, Relationships between Input-Output and State-Space Descriptions, State-Space Analysis in the Z-Domain, Additional State-Space Structures.

Representation of Numbers: Fixed-Point Representation of Numbers, Binary Floating-Point Representation of Numbers, Errors Resulting from Rounding and Truncation.

PART – B

UNIT – 5 **6 Hours**

Implementation of Discrete-Time Systems – 3: Quantization of Filter Coefficients: Analysis of Sensitivity to Quantizatior of Filter Coefficients, Quantization of Coefficients in FIR Filters

Round-Off Effects in Digital Filters: Limit-Cycle Oscillations in Recursive Systems, Scaling to Prevent Overflow, Statistical Characterization of Quantization effects in Fixed-Point Realizations of Digital Filters.

UNIT – 6 **7 Hours**

Design of Digital Filters – 1: General Considerations: Causality and its Implications, Characteristics of Practical Frequency-Selective Filters.

Design of FIR Filters: Symmetric And Antisymetric FIR Filters, Design of Linear-Phase FIR Filters Using Windows, Design of Linear-Phase FIR Filters by the Frequency-Sampling Method, Design of Optimum Equiripple Linear-Phase FIR Filters, Design of FIR Differentiators, Design of Hilbert Transformers, Comparison of Design Methods for Linear-Phase FIR filters.

UNIT – 7 **6 Hours**

Design of Digital Filters – 2: Design of IIR Filters from Analog Filters: IIR Filter Design by Approximation of Derivatives, IIR Filter Design by Impulse Invariance, IIR Filter Design by the Bilinear Transformation, The Matched-Z Transformation, Characteristics of commonly used Analog Filters, Some examples of Digital Filters Designs based on the Bilinear Transformation.

UNIT – 8 **7 Hours**

Design of Digital Filters – 3: Frequency Transformations: Frequency Transformations in the Analog Domain, Frequency Transformations in the Digital Domain.

Design of Digital Filters based on Least-Squares method: Padé Approximations method, Least-Square design methods, FIR least-Squares Inverse (Wiener) Filters, Design of IIR Filters in the Frequency domain.

Text Books:

1. John G. Proakis and Dimitris G. Manolakis: Digital Signal Processing, 3rd Edition, Pearson Education, 2003. (Chapters 5, 6, 7 and 8)

Reference Books:

1. Paulo S. R. Diniz, Eduardo A. B. da Silva And Sergio L. Netto: Digital Signal Processing: System Analysis and Design, Cambridge University Press, 2002.
2. Sanjit K. Mitra: Digital Signal Processing: A Computer Based Approach, Tata Mcgraw-Hill, 2001.
3. Alan V Oppenheim and Ronald W Schafer: Digital Signal Processing, PHI, Indian Reprint, 2008.

JAVA AND J2EE

Subject Code:10CS753	**IA Marks: 25**
Hours/Week: 4	**Exam Marks: 100**
Total Hours: 52	**Exam Hours: 3**

PART - A

UNIT – 1 **6 Hours**

Introduction to Java: Java and Java applications; Java Development Kit (JDK); Java is interpreted, Byte Code, JVM; Object-oriented programming; Simple Java programs.

Data types and other tokens: Boolean variables, int, long, char, operators, arrays, white spaces, literals, assigning values; Creating and destroying objects; Access specifiers.

Operators and Expressions: Arithmetic Operators, Bitwise operators, Relational operators, The Assignment Operator, The ? Operator; Operator Precedence; Logical expression; Type casting; Strings

Control Statements: Selection statements, iteration statements, Jump Statements.

UNIT – 2 **6 Hours**

Classes, Inheritance, Exceptions, Applets : Classes: Classes in Java; Declaring a class; Class name; Super classes; Constructors; Creating instances of class; Inner classes.

Inheritance: Simple, multiple, and multilevel inheritance; Overriding, overloading.

Exception handling**:** Exception handling in Java.

The Applet Class: Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting; Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface; Output to the Console.

UNIT – 3 **7 Hours**

Multi Threaded Programming, Event Handling: Multi Threaded Programming: What are threads? How to make the classes threadable; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem, producer-consumer problems.

Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes.

UNIT – 4 **7 Hours**

Swings: Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable.

PART – B

UNIT – 5 **6 Hours**

Java 2 Enterprise Edition Overview, Database Access: Overview of J2EE and J2SE

The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.

UNIT – 6 **7 Hours**

Servlets: Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking.

UNIT – 7 **6 Hours**

JSP, RMI: Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects.

Java Remote Method Invocation: Remote Method Invocation concept; Server side, Client side.

UNIT – 8 **7 Hours**

Enterprise Java Beans: Enterprise java Beans; Deployment Descriptors; Session Java Bean, Entity Java Bean; Message-Driven Bean; The JAR File.

Text Books:

1. Herbert Schildt: Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.
   (Chapters 1, 2, 3, 4, 5, 6, 8, 10, 11, 21, 22, 29, 30, 31)
2. Jim Keogh: J2EE - The Complete Reference, Tata McGraw Hill, 2007.
   (Chapters 5, 6, 11, 12, 15)

Reference Books:

1. Y. Daniel Liang: Introduction to JAVA Programming, 7th Edition, Pearson Education, 2007.
2. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education, 2004.

MULTIMEDIA COMPUTING

Subject Code: 10CS754	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **7 Hours**

Introduction, Media and Data Streams, Audio Technology: Multimedia Elements; Multimedia Applications; Multimedia Systems Architecture; Evolving Technologies for Multimedia Systems; Defining Objects for Multimedia Systems; Multimedia Data Interface Standards; The need for Data Compression; Multimedia Databases.

Media: Perception Media, Representation Media, Presentation Media, Storage Media, Transmission Media, Information Exchange Media, Presentation Spaces & Values, and Presentation Dimensions; Key Properties of a Multimedia System: Discrete & Continuous Media, Independence Media, Computer Controlled Systems, Integration; Characterizing Data Streams: Asynchronous Transmission Mode, Synchronous Transmission Mode, Isochronous Transmission Mode; Characterizing Continuous Media Data Streams.

Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics; Audio Representation on Computers; Three Dimensional Sound Projection; Music and MIDI Standards; Speech Signals; Speech Output; Speech Input; Speech Transmission.

UNIT – 2 **7 Hours**

Graphics and Images, Video Technology, Computer-Based Animation: Capturing Graphics and Images Computer Assisted Graphics and Image Processing; Reconstructing Images; Graphics and Image Output Options.

Basics; Television Systems; Digitalization of Video Signals; Digital Television; Basic Concepts; Specification of Animations; Methods of Controlling Animation; Display of Animation; Transmission of Animation; Virtual Reality Modeling Language.

UNIT – 3 **7 Hours**

Data Compression – 1: Storage Space; Coding Requirements; Source, Entropy, and Hybrid Coding; Basic Compression Techniques; JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Lossless Mode, Hierarchical Mode

UNIT – 4 **6 Hours**

Data Compression – 2: H.261 (Px64) and H.263: Image Preparation, Coding Algorithms, Data Stream, H.263+ and H.263L; MPEG: Video Encoding, Audio Coding, Data Stream, MPEG-2, MPEG-4, MPEG-7; Fractal Compression.

PART - B

UNIT – 5 **6 Hours**

Optical Storage Media: History of Optical Storage; Basic Technology; Video Discs and Other WORMs; Compact Disc Digital Audio; Compact Disc Read Only Memory; CD-ROM Extended Architecture; Further CD-ROM-Based Developments; Compact Disc Recordable; Compact Disc Magneto-Optical; Compact Disc Read/Write; Digital Versatile Disc.

UNIT – 6 **6 Hours**

Content Analysis : Simple Vs. Complex Features; Analysis of Individual Images; Analysis of Image Sequences; Audio Analysis; Applications.

UNIT – 7 **6 Hours**

Data and File Format Standards: Rich-Text Format; TIFF File Format; Resource Interchange File Format (RIFF); MIDI File Format; JPEG DIB File Format for Still and Motion Images; AVI Indeo File Format; MPEG Standards; TWAIN

UNIT – 8 **7 Hours**

Multimedia Application Design : Multimedia Application Classes; Types of Multimedia Systems; Virtual Reality Design; Components of Multimedia Systems; Organizing Multimedia Databases; Application Workflow Design Issues; Distributed Application Design Issues.

Text Books:

1. Ralf Steinmetz, Klara Narstedt: Multimedia Fundamentals: Vol 1-Media Coding and Content Processing, 2nd Edition, PHI, Indian Reprint 2008.
   (Chapters 2, 3, 4, 5, 6, 7, 8, 9)
2. Prabhat K. Andleigh, Kiran Thakrar: Multimedia Systems Design, PHI, 2003.
   (Chapters 1, 3, 7)

Reference Books:

1. K.R Rao, Zoran S. Bojkovic and Dragorad A. Milovanovic: Multimedia Communication Systems: Techniques, Standards, and Networks, Pearson Education, 2002.
2. Nalin K Sharad: Multimedia Information Networking, PHI, 2002.

DATA WAREHOUSING AND DATA MINING

Subject Code: 10CS755 **I.A. Marks : 25**
Hours/Week : 04 **Exam Hours: 03**
Total Hours : 52 **Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Data Warehousing:
Introduction, Operational Data Stores (ODS), Extraction Transformation Loading (ETL), Data Warehouses. Design Issues, Guidelines for Data Warehouse Implementation, Data Warehouse Metadata

UNIT – 2

6 Hours

Online Analytical Processing (OLAP): Introduction, Characteristics of OLAP systems, Multidimensional view and Data cube, Data Cube Implementations, Data Cube operations, Implementation of OLAP and overview on OLAP Softwares.

UNIT – 3 **6 Hours**

Data Mining: Introduction, Challenges, Data Mining Tasks, Types of Data, Data Preprocessing, Measures of Similarity and Dissimilarity, Data Mining Applications

UNIT – 4 **8 Hours**

Association Analysis: Basic Concepts and Algorithms: Frequent Itemset Generation, Rule Generation, Compact Representation of Frequent Itemsets, Alternative methods for generating Frequent Itemsets, FP Growth Algorithm, Evaluation of Association Patterns

PART - B

UNIT – 5 **6 Hours**

Classification -1 : Basics, General approach to solve classification problem, Decision Trees, Rule Based Classifiers, Nearest Neighbor Classifiers.

UNIT – 6 **6 Hours**

Classification - 2 : Bayesian Classifiers, Estimating Predictive accuracy of classification methods, Improving accuracy of clarification methods, Evaluation criteria for classification methods, Multiclass Problem.

UNIT – 7 **8 Hours**

Clustering Techniques: Overview, Features of cluster analysis, Types of Data and Computing Distance, Types of Cluster Analysis Methods, Partitional Methods, Hierarchical Methods, Density Based Methods, Quality and Validity of Cluster Analysis

UNIT – 8 **6 Hours**

Web Mining: Introduction, Web content mining, Text Mining, Unstructured Text, Text clustering, Mining Spatial and Temporal Databases.

Text Books:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson Education, 2005.
2. G. K. Gupta: Introduction to Data Mining with Case Studies, 3rd Edition, PHI, New Delhi, 2009.

Reference Books:

1. Arun K Pujari: Data Mining Techniques 2nd Edition, Universities Press, 2009.
2. Jiawei Han and Micheline Kamber: Data Mining - Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publisher, 2006.
3. Alex Berson and Stephen J. Smith: Data Warehousing, Data Mining, and OLAP Computing, Mc GrawHill Publisher, 1997.

NEURAL NETWORKS

Subject Code: 10CS756 **I.A. Marks : 25**
Hours/Week : 04 **Exam Hours: 03**
Total Hours : 52 **Exam Marks: 100**

PART – A

UNIT – 1 **7 Hours**

Introduction

What is a Neural Network?, Human Brain, Models of Neuron, Neural Networks viewed as directed graphs, Feedback, Network Architectures, Knowledge representation, Artificial Intelligence and Neural Networks.

UNIT – 2 **6 Hours**

Learning Processes – 1

Introduction, Error-correction learning, Memory-based learning, Hebbian learning, Competitive learning,Boltzamann learning, Credit Assignment problem, Learning with a Teacher, Learning without a Teacher, Learning tasks, Memory, Adaptation.

UNIT – 3 **7 Hours**

Learning Processes – 2, **Single Layer Perceptrons:** Statistical nature of the learning process, Statistical learning theory, Approximately correct model of learning.
Single Layer Perceptrons: Introduction, Adaptive filtering problem, Unconstrained optimization techniques, Linear least-squares filters**,** Least-mean square algorithm, Learning curves, Learning rate annealing techniques, Perceptron, Perceptron convergence theorem, Relation between the Perceptron and Bayes classifier for a Gaussian environment.

UNIT – 4 **6 Hours**
Multilayer Perceptrons – 1: Introduction, Some preliminaries, Back-propagation Algorithm, Summary of back-propagation algorithm, XOR problem, Heuristics for making the back-propagation algorithm perform better, Output representation and decision rule, Computer experiment, Feature detection, Back-propagation and differentiation.

PART - B

UNIT – 5 **7 Hours**
Multilayer Perceptrons – 2: Hessian matrix, Generalization, approximation of functions, Cross validation, Network pruning techniques, virtues and limitations of back- propagation learning, Accelerated convergence of back propagation learning, Supervised learning viewed as an optimization problem, Convolution networks.

UNIT – 6 **6 Hours**
Radial-Basic Function Networks – 1: Introduction, Cover‟s theorem on the separability of patterns, Interpolation problem, Supervised learning as an ill-posed Hypersurface reconstruction problem, Regularization theory, Regularization networks, Generalized radial-basis function networks, XOR problem, Estimation of the regularization parameter.

UNIT – 7 **6 Hours**
Radial-Basic Function Networks – 2, Optimization – 1: Approximation properties of RBF networks, Comparison of RBF networks and multilayer Perceptrons, Kernel regression and it‟s relation to RBF networks, Learning strategies, Computer experiment.
Optimization using Hopfield networks: Traveling salesperson problem, Solving simultaneous linear equations, Allocating documents to multiprocessors.

UNIT – 8 **7 Hours**

Optimization Methods – 2:

Iterated gradient descent, Simulated Annealing, Random Search, Evolutionary computation- Evolutionary algorithms, Initialization, Termination criterion, Reproduction, Operators, Replacement, Schema theorem

Text Books:

1. Simon Haykin: Neural Networks - A Comprehensive Foundation, 2nd Edition, Pearson Education, 1999.
   (Chapters 1.1-1.8, 2.1-2.15, 3.1-3.10, 4.1-4.19, 5.1-5.14)
2. Kishan Mehrotra, Chilkuri K. Mohan, Sanjay Ranka: Artificial Neural Networks, Penram International Publishing, 1997.
   (Chapters 7.1-7.5)

Reference Books:

1. B.Yegnanarayana: Artificial Neural Networks, PHI, 2001.

C# PROGRAMMING AND .NET

Subject Code: 10CS761	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

The philosophy of .NET: Understanding the Previous State of Affairs, The .NET Solution, The Building Block of the .NET Platform (CLR,CTS, and CLS), The Role of the .NET Base Class Libraries, What C# Brings to the Table, An Overview of .NET Binaries (aka Assemblies), the Role of the Common Intermediate Language , The Role of .NET Type Metadata, The Role of the Assembly Manifast, Compiling CIL to Platform –Specific Instructions, Understanding the Common Type System, Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime

UNIT – 2 **6 Hours**

Building C# Applications: The Role of the Command Line Complier (csc.exe), Building C # Application using csc.exe Working with csc.exe Response Files, Generating Bug Reports , Remaining C# Compiler Options, The Command Line Debugger (cordbg.exe) Using the, Visual Studio .NET

IDE, Other Key Aspects of the VS.NET IDE, C# "Preprocessor:" Directives, An Interesting Aside: The System. Environment Class

UNIT – 3 **8 Hours**

C# Language Fundamentals: The Anatomy of a Basic C# Class, Creating objects: Constructor Basics, The Composition of a C# Application, Default Assignment and Variable Scope, The C# Member Initialization Syntax, Basic Input and Output with the Console Class, Understanding Value Types and Reference Types, The Master Node: System, Object, The System Data Types (and C# Aliases), Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs, The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies, Array Manipulation in C #, String Manipulation in C#, C# Enumerations, Defining Structures in C#, Defining Custom Namespaces

UNIT – 4 **6 Hours**

Object- Oriented Programming with C#: Forms Defining of the C# Class, Definition the "Default Public Interface" of a Type, Recapping the Pillars of OOP, The First Pillars: C#"s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields, The Second Pillar: C#"s Inheritance Supports, keeping Family Secrets: The "Protected" Keyword, Nested Type Definitions, The Third Pillar: C #"s Polymorphic Support, Casting Between.

PART – B

UNIT – 5 **6 Hours**

Exceptions and Object Lifetime: Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handing, the System. Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception(System. System Exception), Custom Application-Level Exception(System. System Exception), Handling Multiple Exception, The Family Block, the Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the CIT of "new", The Basics of Garbage Collection,, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type.

UNIT – 6 **6 Hours**

Interfaces and Collections: Defining Interfaces Using C# Invoking Interface Members at the object Level, Exercising the Shapes Hierarchy, Understanding Explicit Interface Implementation, Interfaces As Polymorphic Agents, Building Interface Hierarchies, Implementing, Implementation, Interfaces Using VS .NET, understanding the IConvertible Interface, Building a Custom Enumerator (IEnumerable and Enumerator), Building Cloneable objects (ICloneable), Building Comparable Objects (I Comparable), Exploring the system. Collections Namespace, Building a Custom Container (Retrofitting the Cars Type)

UNIT – 7 **8 Hours**

Callback Interfaces, Delegates, and Events, Advanced Techniques: Understanding Callback Interfaces, Understanding the .NET Delegate Type, Members of System. Multicast Delegate, The Simplest Possible Delegate Example, Building More a Elaborate Delegate Example, Understanding Asynchronous Delegates, Understanding (and Using)Events.

The Advances Keywords of C#, A Catalog of C# Keywords Building a Custom Indexer, A Variation of the Cars Indexer Internal Representation of Type Indexer . Using C# Indexer from VB .NET. Overloading operators, The Internal Representation of Overloading Operators, interacting with Overload Operator from Overloaded- Operator- Challenged Languages, Creating Custom Conversion Routines, Defining Implicit Conversion Routines, The Internal Representations of Customs Conversion Routines

UNIT – 8 **6 Hours**

Understanding .NET Assembles: Problems with Classic COM Binaries, An Overview of .NET Assembly, Building a Simple File Test Assembly, A C#. Client Application, A Visual Basic .NET Client Application, Cross Language Inheritance, Exploring the CarLibrary‟s, Manifest, Exploring the CarLibrary‟s Types, Building the Multifile Assembly, Using Assembly, Understanding Private Assemblies, Probing for Private Assemblies (The Basics), Private A Assemblies XML Configurations Files, Probing for Private Assemblies (The Details), Understanding Shared Assembly, Understanding Shared Names, Building a Shared Assembly, Understanding Delay Signing, Installing/Removing Shared Assembly, Using a Shared Assembly

Text Books:

1. Andrew Troelsen: Pro C# with .NET 3.0, 4th Edition, Wiley India, 2009.
   Chapters: 1 to 11 (up to pp.369)
2. E. Balagurusamy: Programming in C#, 2nd Edition, Tata McGraw Hill, 2008.

(Programming Examples 3.7, 3.10, 5.5, 6.1, 7.2, 7.4, 7.5, 7.6, 8.1, 8.2, 8.3, 8.5, 8.7, 8.8, 9.1, 9.2, 9.3, 9.4, 10.2, 10.4, 11.2, 11.4, 12.1, 12.4, 12.5, 12.6, 13.1, 13.2, 13.3, 13.6, 14.1, 14.2, 14.4, 15.2, 15.3, 16.1, 16.2, 16.3, 18.3, 18.5.18.6)

Reference Books:

1. Tom Archer: Inside C#, WP Publishers, 2001.
2. Herbert Schildt: C# The Complete Reference, Tata McGraw Hill, 2004.

DIGITAL IMAGE PROCESSING

Subject Code: 10CS762	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Digitized Image and its properties: Basic concepts, Image digitization, Digital image properties

UNIT – 2 **7 Hours**

Image Preprocessing: Image pre-processing: Brightness and geometric transformations, local preprocessing.

UNIT – 3 **7 Hours**

Segmentation – 1: Thresholding, Edge-based segmentation.

UNIT – 4 **7 Hours**

Segmentation – 2: Region based segmentation, Matching.

PART – B

UNIT – 5 **7 Hours**

Image Enhancement: Image enhancement in the spatial domain: Background, Some basic gray level transformations, Histogram processing, Enhancement using arithmetic/ logic operations, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters. Image enhancement in the frequency domain: Background, Introduction to the Fourier transform and the frequency domain, Smoothing Frequency-Domain filters, Sharpening Frequency Domain filters, Homomorphic filtering.

UNIT – 6 **6 Hours**

Image Compression: Image compression: Fundamentals, Image compression models, Elements of information theory, Error-Free Compression, Lossy compression.

UNIT – 7 **7 Hours**

Shape representation: Region identification, Contour-based shape representation and description, Region based shape representation and description, Shape classes.

UNIT – 8 **6 Hours**

Morphology: Basic morphological concepts, Morphology principles, Binary dilation and erosion, Gray-scale dilation and erosion, Morphological segmentation and watersheds

Text Books:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle: Image Processing, Analysis and Machine Vision, 2nd Edition, Thomoson Learning, 2001.
   (Chapters 2, 4.1 to 4.3, 5.1 to 5.4, 6, 11.1 to 11.4, 11.7)
2. Rafel C Gonzalez and Richard E Woods: Digital Image Processing, 3rd Edition, Pearson Education, 2003.
   (Chapters 3.1 to 3.7, 4.1 to 4.5, 8.1 to 8.5)

Reference Books:

1. Anil K Jain, "Fundamentals of Digital Image Processing", PHI, 1997, Indian Reprint 2009.
2. B.Chanda, D Dutta Majumder, "Digital Image Processing and Analysis", PHI, 2002.

GAME THEORY

Subject Code: 10CS763	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT – 1 **8 Hours**

Introduction, Strategic Games: What is game theory? The theory of rational choice; Interacting decision makers.

Strategic games; Examples: The prisoner‟s dilemma, Bach or Stravinsky, Matching pennies; Nash equilibrium; Examples of Nash equilibrium; Best-response functions; Dominated actions; Equilibrium in a single population: symmetric games and symmetric equilibria.

UNIT – 2 **6 Hours**

Mixed Strategy Equilibrium: Introduction; Strategic games in which players may randomize; Mixed strategy Nash equilibrium; Dominated actions; Pure equilibria when randomization is allowed, Illustration: Expert Diagnosis; Equilibrium in a single population, Illustration: Reporting a crime; The formation of players‟ beliefs; Extensions; Representing preferences by expected payoffs.

UNIT – 3 **6 Hours**

Extensive Games: Extensive games with perfect information; Strategies and outcomes; Nash equilibrium; Subgame perfect equilibrium; Finding subgame perfect equilibria of finite horizon games: Backward induction. Illustrations: The ultimatum game, Stackelberg‟s model of duopoly, Buying votes.

UNIT – 4 **6 Hours**

Extensive games: Extensions and Discussions: Extensions: Allowing for simultaneous moves, Illustrations: Entry in to a monopolized industry, Electoral competition with strategic voters, Committee decision making, Exit from a declining industry; Allowing for exogenous uncertainty, Discussion: subgame perfect equilibrium and backward induction.

PART – B

UNIT – 5 **7 Hours**

Bayesian Games, Extensive Games with Imperfect Information: Motivational examples; General definitions; Two examples concerning information; Illustrations: Cournot‟s duopoly game with imperfect information, Providing a public good, Auctions; Auctions with an arbitrary distribution of valuations.

Extensive games with imperfect information; Strategies; Nash equilibrium; Beliefs and sequential equilibrium; Signaling games; Illustration: Strategic information transmission.

UNIT – 6 **7 Hours**

Strictly Competitive Games, Evolutionary Equilibrium: Strictly competitive games and maximization; Maximization and Nash equilibrium; Strictly competitive games; Maximization and Nash equilibrium in strictly competitive games.

Evolutionary Equilibrium: Monomorphic pure strategy equilibrium; Mixed strategies and polymorphic equilibrium; Asymmetric contests; Variations on themes: Sibling behavior, Nesting behavior of wasps, The evolution of sex ratio.

UNIT – 7 **6 Hours**

Iterated Games: Repeated games: The main idea; Preferences; Repeated games; Finitely and infinitely repeated Prisoner‟s dilemma; Strategies in an infinitely repeated Prisoner‟s dilemma; Some Nash equilibria of an infinitely repeated Prisoner‟s dilemma, Nash equilibrium payoffs of an infinitely repeated Prisoner‟s dilemma.

UNIT – 8 **6 Hours**

Coalitional Games and Bargaining: Coalitional games. The Core. Illustrations: Ownership and distribution of wealth, Exchanging homogeneous items, Exchanging heterogeneous items, Voting, Matching. Bargaining as an extensive game; Illustration of trade in a market; Nash's axiomatic model of bargaining

Text Books:

1. Martin Osborne: An Introduction to Game Theory, Oxford University Press, Indian Edition, 2004.
   (Listed topics only from Chapters 1 to 11, 13, 14, 16)

Reference Books:

1. Roger B. Myerson: Game Theory: Analysis of Conflict, Harvard University Press, 1997.
2. Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green: Microeconomic Theory. Oxford University Press, New York, 1995.
3. Philip D. Straffin, Jr.: Game Theory and Strategy, The Mathematical Association of America, January 1993.

ARTIFICIAL INTELLIGENCE

Subject Code: 10CS764	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **7 Hours**

Introduction: What is AI? Intelligent Agents: Agents and environment; Rationality; the nature of environment; the structure of agents. Problem-solving: Problem-solving agents; Example problems; Searching for solution; Uninformed search strategies.

UNIT – 2 **7 Hours**

Informed Search, Exploration, Constraint Satisfaction, Adversial Search: Informed search strategies; Heuristic functions; On-line search agents and unknown environment. Constraint satisfaction problems; Backtracking search

for CSPs. Adversial search: Games; Optimal decisions in games; Alpha-Beta pruning.

UNIT – 3 **6 Hours**

Logical Agents: Knowledge-based agents; The wumpus world as an example world; Logic; propositional logic Reasoning patterns in propositional logic; Effective propositional inference; Agents based on propositional logic.

UNIT – 4 **6 Hours**

First-Order Logic, Inference in First-Order Logic – 1: Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Propositional versus first-order inference; Unification and lifting

PART – B

UNIT – 5 **6 Hours**

Inference in First-Order Logic – 2: Forward chaining; Backward chaining; Resolution.

UNIT – 6 **7 Hours**

Knowledge Representation: Ontological engineering; Categories and objects; Actions, situations, and events; Mental events and mental objects; The Internet shopping world; Reasoning systems for categories; Reasoning with default information; Truth maintenance systems.

UNIT – 7 **7 Hours**

Planning, Uncertainty, Probabilistic Reasoning: Planning: The problem; Planning with state-space approach; Planning graphs; Planning with propositional logic.

Uncertainty: Acting under certainty; Inference using full joint distributions; Independence; Bayes‟ rule and its use.

Probabilistic Reasoning: Representing knowledge in an uncertain domain; The semantics of Bayesian networks; Efficient representation of conditional distributions; Exact inference in Bayesian networks.

UNIT – 8 **6 Hours**

Learning, AI: Present and Future: Learning: Forms of Learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory.

AI: Present and Future: Agent components; Agent architectures; Are we going in the right direction? What if AI does succeed?

Text Books:

1. Stuart Russel, Peter Norvig: Artificial Intelligence A Modern Approach, 2nd Edition, Pearson Education, 2003.
(Chapters 1.1, 2, 3.1 to 3.4, 4.1, 4.2, 4.5, 5.1, 5.2, 6.1, 6.2, 6.3, 7, 8, 9, 10, 11.1, 11.2, 11.4, 11.5, 13.1, 13.4, 13.5, 13.6, 14.1, 14.2, 14.3, 14.4, 18, 27)

Reference Books:

1. Elaine Rich, Kevin Knight: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2009.
2. Nils J. Nilsson: Principles of Artificial Intelligence, Elsevier, 1980.

STORAGE AREA NETWORKS

Subject Code: 10CS765	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART –A

UNIT - 1 **7 Hours**

Introduction to Information Storage and Management, Storage System Environment: Information Storage, Evolution of Storage Technology and Architecture, Data Center Infrastructure, Key Challenges in Managing Information, Information Lifecycle

Components of Storage System Environment, Disk Drive Components, Disk Drive Performance, Fundamental Laws Governing Disk Performance, Logical Components of the Host, Application Requirements and Disk Performance.

UNIT - 2 **6 Hours**

Data Protection, Intelligent Storage system: Implementation of RAID, RAID Array Components, RAID Levels, RAID Comparison, RAID Impact on Disk Performance, Hot Spares

Components of an Intelligent Storage System, Intelligent Storage Array

UNIT - 3 **7 Hours**

Direct-Attached Storage, SCSI, and Storage Area Networks: Types of DAS, DAS Benefits and Limitations, Disk Drive Interfaces, Introduction to Parallel SCSI, Overview of Fibre Channel, The SAN and Its Evolution, Components of SAN, FC Connectivity, Fibre Channel Ports, Fibre Channel Architecture, Zoning, Fibre Channel Login Types, FC Topologies.

UNIT - 4 **6 Hours**

NAS, IP SAN: General – Purpose Service vs. NAS Devices, Benefits of NAS, NAS File I / O, Components of NAS, NAS Implementations, NAS

File-Sharing Protocols, NAS I/O Operations, Factors Affecting NAS Performance and Availability. iSCSI, FCIP.

PART - B

UNIT - 5 **6 Hours**

Content-Addressed Storage, Storage Virtualization: Fixed Content and Archives, Types of Archive, Features and Benefits of CAS, CAS Architecture, Object Storage and Retrieval in CAS, CAS Examples

Forms of Virtualization, SNIA Storage Virtualization Taxonomy, Storage Virtualizations Configurations, Storage Virtualization Challenges, Types of Storage Virtualization

UNIT - 6 **6 Hours**

Business Continuity, Backup and Recovery: Information Availability, BC Terminology, BC Planning Lifecycle, Failure Analysis, Business Impact Analysis, BC Technology Solutions.

Backup Purpose, Backup Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup Process, Backup and restore Operations, Backup Topologies, Backup in NAS Environments, Backup Technologies.

UNIT - 7 **7 Hours**

Local Replication, Remote Replication: Source and Target, Uses of Local Replicas, Data Consistency, Local Replication Technologies, Restore and Restart Considerations, Creating Multiple Replicas, Management Interface, Modes of Remote Replication, Remote Replication Technologies, Network Infrastructure.

UNIT - 8 **7 Hours**

Securing the Storage Infrastructure, Managing the Storage Infrastructure: Storage Security Framework, Risk Triad, Storage Security Domains, Security Implementations in Storage Networking

Monitoring the Storage Infrastructure, Storage Management Activities, Storage Infrastructure Management Challenges, Developing an Ideal Solution.

Text Books:

1. G. Somasundaram, Alok Shrivastava (Editors): Information Storage and Management, EMC Education Services, Wiley India, 2009.

Reference Books:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2003.
2. Rebert Spalding: Storage Networks, The Complete Reference, Tata McGraw Hill, 2003.

3. Richard Barker and Paul Massiglia: Storage Area Networks Essentials A Complete Guide to Understanding and Implementing SANs, Wiley India, 2002.

FUZZY LOGIC

Subject Code: 10CS766	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **7 Hours**

Introduction, Classical Sets and Fuzzy Sets: Background, Uncertainty and Imprecision, Statistics and Random Processes, Uncertainty in Information, Fuzzy Sets and Membership, Chance versus Ambiguity.

Classical Sets - Operations on Classical Sets, Properties of Classical (Crisp) Sets, Mapping of Classical Sets to Functions

Fuzzy Sets - Fuzzy Set operations, Properties of Fuzzy Sets. Sets as Points in Hypercubes

UNIT – 2 **6 Hours**

Classical Relations and Fuzzy Relations: Cartesian Product, Crisp Relations - Cardinality of Crisp Relations, Operations on Crisp Relations, Properties of Crisp Relations, Composition. Fuzzy Relations - Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Cartesian Product and Composition, Non-interactive Fuzzy Sets. Tolerance and Equivalence Relations - Crisp Equivalence Relation, Crisp Tolerance Relation, Fuzzy Tolerance and Equivalence Relations. Value Assignments - Cosine Amplitude, Max-min Method, Other Similarity methods

UNIT – 3 **6 Hours**

Membership Functions: Features of the Membership Function, Standard Forms and Boundaries, Fuzzification, Membership Value Assignments – Intuition, Inference, Rank Ordering, Angular Fuzzy Sets, Neural Networks, Genetic Algorithms, Inductive Reasoning.

UNIT – 4 **7 Hours**

Fuzzy-to-Crisp Conversions, Fuzzy Arithmetic: Lambda-Cuts for Fuzzy Sets, Lambda-Cuts for Fuzzy Relations, Defuzzification Methods

Extension Principle - Crisp Functions, Mapping and Relations, Functions of fuzzy Sets – Extension Principle, Fuzzy Transform (Mapping), Practical Considerations, Fuzzy Numbers

Interval Analysis in Arithmetic, Approximate Methods of Extension - Vertex method, DSW Algorithm, Restricted DSW Algorithm, Comparisons, Fuzzy Vectors

PART - B

UNIT – 5 **6 Hours**

Classical Logic and Fuzzy Logic: Classical Predicate Logic – Tautologies, Contradictions, Equivalence, Exclusive OR and Exclusive NOR, Logical Proofs, Deductive Inferences. Fuzzy Logic, Approximate Reasoning, Fuzzy Tautologies, Contradictions, Equivalence and Logical Proofs, Other forms of the Implication Operation, Other forms of the Composition Operation

UNIT – 6 **6 Hours**

Fuzzy Rule- Based Systems: Natural Language, Linguistic Hedges, Rule-Based Systems - Canonical Rule Forms, Decomposition of Compound Rules, Likelihood and Truth Qualification, Aggregation of Fuzzy Rules, Graphical Techniques of Inference

UNIT – 7 **7 Hours**

Fuzzy Decision Making : Fuzzy Synthetic Evaluation, Fuzzy Ordering, Preference and consensus, Multiobjective Decision Making, Fuzzy Bayesian Decision Method, Decision Making under Fuzzy States and Fuzzy Actions.

UNIT – 8 **7 Hours**

Fuzzy Classification: Classification by Equivalence Relations - Crisp Relations, Fuzzy Relations. Cluster Analysis, Cluster Validity, c-Means Clustering - Hard c-Means (HCM), Fuzzy c-Means (FCM). Classification Metric, Hardening the Fuzzy c-Partition, Similarity Relations from Clustering

Text Books:

1. Timothy J. Ross: Fuzzy Logic with Engineering Applications, 2nd Edition, Wiley India, 2006..
(Chapter 1 (pp 1-14), Chapter 2 (pp 17-34), Chapter 3 (pp 46-70), Chapter 4 (pp 87-122), Chapter 5 (pp 130-146), Chapter 6 (pp 151-178), Chapter 7 (pp 183-210), Chapter 8 (pp 232-254), Chapter 9 (pp 313-352), Chapter 10 (pp 371 – 400))

Reference Books:

1. B Kosko: Neural Networks and Fuzzy systems: A Dynamical System approach, PHI, 1991.

Networks Laboratory

Subject Code: 10CSL77	**I.A. Marks : 25**
Hours/Week : 03	**Exam Hours: 03**
Total Hours : 42	**Exam Marks: 50**

Note: Student is required to solve one problem from PART-A and one problem from PART-B. The questions are allotted based on lots. Both questions carry equal marks.

PART A – Simulation Exercises

The following experiments shall be conducted using either NS228/OPNET or any other suitable simulator.

1. Simulate a three nodes point – to – point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.
2. Simulate a four node point-to-point network with the links connected as follows:
   n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0-n3 and UDP between n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP / UDP.
3. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
4. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare throughput.
5. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
6. Simulate simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

PART-B

Implement the following in C/C++:

7. Write a program for error detecting code using CRC-CCITT (16- bits).
8. Write a program for distance vector algorithm to find suitable path for transmission.
9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.

10. Implement the above program using as message queues or FIFOs as IPC channels.
11. Write a program for simple RSA algorithm to encrypt and decrypt the data.
12. Write a program for congestion control using leaky bucket algorithm.

Note:
In the examination, a combination of one problem has to be asked from Part A for a total of 25 marks and one problem from Part B has to be asked for a total of 25 marks. The choice must be based on random selection from the entire lots.

Web Programming Laboratory

Subject Code: 10CSL78	**I.A. Marks : 25**
Hours/Week : 03	**Exam Hours: 03**
Total Hours : 42	**Exam Marks: 50**

1. Develop and demonstrate a XHTML file that includes Javascript script for the following problems:
   a) Input: A number n obtained using prompt
      Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
      Output: A table of numbers from 1 to n and their squares using **alert**
2. a) Develop and demonstrate, using Javascript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
   b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)
3. a) Develop and demonstrate, using Javascript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.
   b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom.
4. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include

USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.

5. a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.

   b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.
6. a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.

   b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
7. Write a Perl program to display a digital clock which displays the current time of the server.
8. Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.
9. Write a PHP program to store current date-time in a COOKIE and display the „Last visited on" date-time on the web page upon reopening of the same page.
10. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.
11. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.
12. Build a Rails application to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

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

VIII SEMESTER

SOFTWARE ARCHITECTURES

Subject Code: 10IS81	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**

Introduction: The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle; What makes a "good" architecture? What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views.

UNIT – 2 **7 Hours**

Architectural Styles and Case Studies: Architectural styles; Pipes and filters; Data abstraction and object-oriented organization; Event-based, implicit invocation; Layered systems; Repositories; Interpreters; Process control; Other familiar architectures; Heterogeneous architectures. Case Studies: Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style.

UNIT – 3 **6 Hours**

Quality: Functionality and architecture; Architecture and quality attributes; System quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architecture qualities. Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Architectural patterns and styles.

UNIT – 4 **7 Hours**

Architectural Patterns – 1: Introduction; From mud to structure: Layers, Pipes and Filters, Blackboard.

PART – B

UNIT – 5 **7 Hours**

Architectural Patterns – 2: Distributed Systems: Broker; Interactive Systems: MVC, Presentation-Abstraction-Control.

UNIT – 6 **6 Hours**

Architectural Patterns – 3: Adaptable Systems: Microkernel; Reflection.

UNIT – 7 **6 Hours**

Some Design Patterns: Structural decomposition: Whole – Part; Organization of work: Master – Slave; Access Control: Proxy.

UNIT – 8 **7 Hours**

Designing and Documenting Software Architecture: Architecture in the life cycle; Designing the architecture; Forming the team structure; Creating a skeletal system. Uses of architectural documentation; Views; Choosing the relevant views; Documenting a view; Documentation across views.

Text Books:

1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson Education, 2003.
   (Chapters 1, 2, 4, 5, 7, 9)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007.
   (Chapters 2, 3.1 to 3.4)
3. Mary Shaw and David Garlan: Software Architecture- Perspectives on an Emerging Discipline, PHI, 2007.
   (Chapters 1.1, 2, 3)

Reference Books:

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Pearson Education, 1995.

Web Reference: http://www.hillside.net/patterns/

SYSTEM MODELING AND SIMULATION

Sub Code: 10CS82		**IA Marks**	**25**
Hrs/Week: 04		**Exam Hours**	**03**
Total Hrs: 52		**Exam Marks**	**100**

PART – A

UNIT – 1 **8 Hours**

Introduction: When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Model of a system; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study. The basics of Spreadsheet simulation, Simulation example: Simulation of queuing systems in a spreadsheet.

UNIT – 2 **6 Hours**

General Principles, Simulation Software: Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling; List processing. Simulation in Java; Simulation in GPSS

UNIT – 3 **6 Hours**

Statistical Models in Simulation: Review of terminology and concepts; Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.

UNIT – 4 **6 Hours**

Queuing Models: Characteristics of queuing systems; Queuing notation; Long-run measures of performance of queuing systems; Steady-state behavior of M/G/1 queue; Networks of queues; Rough-cut modeling: An illustration..

PART – B

UNIT – 5 **8 Hours**

Random-Number Generation, Random-Variate Generation: Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.

UNIT – 6 **6 Hours**

Input Modeling : Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series input models.

UNIT – 7 **6 Hours**

Estimation of Absolute Performance: Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.

UNIT – 8 **6 Hours**

Verification, Calibration, and Validation; Optimization: Model building, verification and validation; Verification of simulation models; Calibration and validation of models, Optimization via Simulation

Text Books:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010.
   (Listed topics only from Chapters1 to 12)

Reference Books:

1. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson Education, 2006.
2. Averill M. Law: Simulation Modeling and Analysis, 4th Edition, Tata McGraw-Hill, 2007.

WIRELESS NETWORKS AND MOBILE COMPUTING

Sub Code: 10CS831	**IA Marks**	**: 25**
Hrs/Week: 04	**Exam Hours**	**: 03**
Total Hrs: 52	**Exam Marks**	**: 100**

PART-A

UNIT – 1 **6 Hours**

Mobile Computing Architecture: Types of Networks, Architecture for Mobile Computing, 3-tier Architecture, Design Considerations for Mobile Computing.

UNIT – 2 **7 Hours**

Wireless Networks – 1: GSM and SMS: Global Systems for Mobile Communication (GSM and Short Service Messages (SMS): GSM Architecture, Entities, Call routing in GSM, PLMN Interface, GSM Addresses and Identities, Network Aspects in GSM, Mobility Management, GSM Frequency allocation. Introduction to SMS, SMS Architecture, SM MT, SM MO, SMS as Information bearer, applications

UNIT – 3 **6 Hours**

Wireless Networks – 2: GPRS : GPRS and Packet Data Network, GPRS Network Architecture, GPRS Network Operations, Data Services in GPRS, Applications for GPRS, Billing and Charging in GPRS

UNIT – 4 **7 Hours**

Wireless Networks – 3: CDMA, 3G and WiMAX: Spread Spectrum technology, IS-95, CDMA versus GSM, Wireless Data, Third Generation Networks, Applications on 3G, Introduction to WiMAX.

PART - B

UNIT – 5 **6 Hours**

Mobile Client: Moving beyond desktop, Mobile handset overview, Mobile phones and their features, PDA, Design Constraints in applications for handheld devices.Mobile IP: Introduction, discovery, Registration, Tunneling, Cellular IP, Mobile IP with IPv6

UNIT – 6 **7 Hours**

Mobile OS and Computing Environment: Smart Client Architecture, The Client: User Interface, Data Storage, Performance, Data Synchronization, Messaging. The Server: Data Synchronization, Enterprise Data Source, Messaging. Mobile Operating Systems: WinCE, Palm OS, Symbian OS, Linux, Proprietary OS Client Development : The development process, Need analysis phase, Design phase, Implementation and Testing phase, Deployment phase, Development Tools, Device Emulators.

UNIT – 7 **6 Hours**

Building, Mobile Internet Applications: Thin client: Architecture, the client, Middleware, messaging Servers, Processing a Wireless request, Wireless Applications Protocol (WAP) Overview, Wireless Languages: Markup Languages, HDML, WML, HTML, cHTML, XHTML, VoiceXML.

UNIT – 8 **7 Hours**

J2ME: Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet

event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP.

Text Books:

1. Dr. Ashok Talukder, Ms Roopa Yavagal, Mr. Hasan Ahmed: Mobile Computing, Technology, Applications and Service Creation, 2d Edition, Tata McGraw Hill, 2010
2. Martyn Mallik: Mobile and Wireless Design Essentials, Wiley, 2003

Reference Books:

1. Raj kamal: Mobile Computing, Oxford University Press, 2007.
2. Iti Saha Misra: Wireless Communications and Networks, 3G and Beyond, Tata McGraw Hill, 2009.

WEB 2.0 AND RICH INTERNET APPLICATIONS

Sub Code: 10CS832	**IA Marks**	**: 25**
Hrs/ Week: 04	**Exam Hours**	**: 03**
Total Hours: 52	**Exam Marks**	**: 100**

PART - A

UNIT – 1 **6 Hours**

Introduction, Ajax – 1: Web 2.0 and Rich Internet Applications, Overview of Ajax, Examples of usage of Ajax: Updating web page text, Chatting in real time, Dragging and dropping, Downloading images. Creating Ajax Applications: An example, Analysis of example ajax.html, Creating the JavaScript, Creating and opening the XMLHttpRequest object, Data download, Displaying the fetched data, Connecting to the server, Adding Server-side programming, Sending data to the server using GET and POST, Using Ajax together with XML.

UNIT – 2 **7 Hours**

Ajax – 2: Handling multiple XMLHttpRequest objects in the same page, Using two XMLHttpRequest objects, Using an array of XMLHttpRequest objects, Using inner functions, Downloading JavaScript, connecting to Google Suggest, Creating google.php, Downloading from other domains with Ajax, HTML header request and Ajax, Defeating caching, Examples.Building XML and working with XML in JavaScript, Getting the document element, Accessing any XML element, Handling whitespace in Firefox, Handling cross-browser whitespace, Accessing XML data directly, Validating XML, Further examples of Rich Internet Applications with Ajax.

UNIT – 3 **6 Hours**

Ajax – 3: Drawing user‟s attention to downloaded text, Styling text, colors and background using CSS, Setting element location in the web pages, Setting the stacking order of web page elements, Further examples of using Ajax. Displaying all the data in an HTML form, Working with PHP server variables, Getting the data in to array format, Wrapping applications in to a single PHP page, Validating input from the user, Validating integers and text, DOM, Appending new elements to a web page using the DOM and Ajax, Replacing elements using the DOM, Handling timeouts in Ajax, Downloading images with Ajax, Example programs.

UNIT – 4 **7 Hours**

Flex – 1 : Introduction: Understanding Flex Application Technologies, Using Flex Elements, Working with Data Services (Loading Data at Runtime), The Differences between Traditional and Flex Web Applications, Understanding How Flex Applications Work, Understanding Flex and Flash Authoring. Building Applications with the Flex Framework: Using Flex Tool Sets, Creating Projects, Building Applications, Deploying Applications Framework Fundamentals: Understanding How Flex Applications Are Structured, Loading and Initializing Flex Applications, Understanding the Component Life Cycles, Loading One Flex Application into Another Flex Application, Differentiating Between Flash Player and the Flex Framework, Caching the Framework, Understanding Application Domains, Localization, Managing Layout: Flex Layout Overview, Making Fluid Interfaces, Putting It All Together.

PART B

UNIT – 5 **7 Hours**

Flex – 2: MXML: Understanding MXML Syntax and Structure, Making MXML Interactive Working with UI Components: Understanding UI Components, Buttons, Value Selectors, Text Components, List-Based Controls, Pop-Up Controls, Navigators, Control Bars Customizing Application Appearance: Using Styles, Skinning components, Customizing the preloader, Themes, Runtime CSS

UNIT – 6 **6 Hours**

Flex – 3: ActionScript: Using ActionScript, MXML and ActionScript Correlations, Understanding ActionScript Syntax, Variables and Properties, Inheritance, Interfaces, Handling Events, Error Handling, Using XML

UNIT – 7 **7 Hours**

Flex – 4: Managing State: Creating States, Applying States, Defining States, Adding and Removing Components, Setting Properties, Setting Styles,

Setting Event Handlers, Using Action Scripts to Define States, Managing Object Creation Policies, Handling State Events, Understanding State Life Cycles, When To Use States.Using Effects and Transitions: Using Effects, Creating Custom Effects, Using Transitions, Creating Custom Transitions.

UNIT – 8 **6 Hours**

Flex – 5: Working with Data: Using Data Models, Data Binding, Enabling Data Binding for Custom Classes, Data Binding Examples, Building data binding proxies.Validating and Formatting Data: Validating user input, Formatting Data.

Text Books:

1. Steven Holzner: Ajax: A Beginner‟s Guide, Tata McGraw Hill, 2009.
   (Listed topics from Chapters 3, 4, 6, 7, 11, 12)
2. Chafic Kazon and Joey Lott: Programming Flex 3, O‟Reilly, June 2009.
   (Listed topics from Chapters 1 to 8, 12 to 15)

Reference Books:

1. Jack Herrington and Emily Kim: Getting Started with Flex 3, O‟Reilly, 1st Edition, 2008.
2. Michele E. Davis and John A. Phillips: Flex 3 - A Beginner‟s Guide, Tata McGraw-Hill, 2008.
3. Colin Moock: Essential Actionscript 3.0, O‟Reilly Publications, 2007.
4. Nicholas C Zakas et al : Professional Ajax, 2nd Edition, Wrox/Wiley India, 2008.

VLSI DESIGN AND ALGORITHMS

Sub Code: 10CS833 **IA Marks 25**
Hrs/Week: 04 **Exam Hours : 03**
Total Hrs: 52 **Exam Marks : 100**

PART - A

UNIT 1 **6 Hours**

Digital Systems and VLSI: Why design Integrated Circuits? Integrated Circuits manufacturing, CMOS Technology, Integrated Circuit Design Techniques, IP-based Design.

UNIT 2 **8 Hours**

Fabrication and Devices: Fabrication Processes, Transistors, Wires and vias, SCMOS Design Rules, Layout design and tools.

UNIT 3 **6 Hours**

Logic Gates – 1: Combinatorial logic functions, Static Complementary gates, Switch Logic.

UNIT 4 **6 Hours**

Logic Gates – 2: Alternative gate Circuits, Low Power gates, Delay through resistive interconnect; Delay through inductive interconnect, Design for yield, Gates as IP.

PART - B

UNIT 5 **6 Hours**

Combinational Logic Networks: Standard cell-based layout, Combinatorial network delay, Logic and interconnect design, Power Optimization, Switch logic networks, Combinational logic testing.

UNIT 6 **6 Hours**

Sequential Machines: Latches and Flip-flops, Sequential systems and clocking disciplines, Clock generators, Sequential systems design, Power optimization, Design validation, Sequential testing.

UNIT 7 **6 Hours**

Architecture Design: Register Transfer design, High Level Synthesis, Architecture for Low Power, Architecture testing.

UNIT 8 **8 Hours**

Design Problems and Algorithms : Placement and Partitioning: Circuit Representation, Wire-length Estimation, Types of Placement Problems, Placement Algorithms, Constructive Placement, Iterative Improvement, Partitioning, The Kernighan-Lin Partitioning Algorithm. Floor Planning: Concepts, Shape functions and floor plan sizing.Routing: Types of Local Routing Problems, Area Routing, Channel Routing, Introduction to Global Routing, Algorithms for Global Routing

Text Books:

1. Wayne Wolf: Modern VLSI Design - IP-Based Design, 4th Edition, PHI Learning, 2009.
   (Listed topics only from Chapters 1 to 5, and 8)
2. Sabih H. Gerez: Algorithms for VLSI Design Automation, Wiley India, 2007.
   (Listed topics only from Chapters 7, 8, and 9)

NETWORK MANAGEMENT SYSTEMS

Sub Code: 10CS834		**IA Marks**	**25**
Hrs/Week: 04		**Exam Hours**	**03**
Total Hrs: 52		**Exam Marks**	**100**

PART – A

UNIT 1 **7 Hours**

Introduction: Analogy of Telephone Network Management, Data and Telecommunication Network Distributed computing Environments, TCP/IP-Based Networks: The Internet and Intranets, Communications Protocols and Standards- Communication Architectures, Protocol Layers and Services; Case Histories of Networking and Management – The Importance of topology , Filtering Does Not Reduce Load on Node, Some Common Network Problems; Challenges of Information Technology Managers, Network Management: Goals, Organization, and Functions- Goal of Network Management, Network Provisioning, Network Operations and the NOC, Network Installation and Maintenance; Network and System Management, Network Management System platform, Current Status and Future of Network Management.

UNIT 2 **6 Hours**

Basic Foundations: Standards, Models, and Language: Network Management Standards, Network Management Model, Organization Model, Information Model – Management Information Trees, Managed Object Perspectives, Communication Model; ASN.1- Terminology, Symbols, and Conventions, Objects and Data Types, Object Names, An Example of ASN.1 from ISO 8824; Encoding Structure; Macros, Functional Model.

UNIT 3 **6 Hours**

SNMPv1 Network Management - 1 : Managed Network: The History of SNMP Management, Internet Organizations and standards, Internet Documents, The SNMP Model, The Organization Model, System Overview.

UNIT 4 **7 Hours**

SNMPv1 Network Management – 2: The Information Model – Introduction, The Structure of Management Information, Managed Objects, Management Information Base.The SNMP Communication Model – The SNMP Architecture, Administrative Model, SNMP Specifications, SNMP Operations, SNMP MIB Group, Functional Model

PART - B

UNIT 5 **6 Hours**

SNMP Management – RMON: Remote Monitoring, RMON SMI and MIB, RMONI1- RMON1 Textual Conventions, RMON1 Groups and Functions, Relationship Between Control and Data Tables, RMON1 Common and Ethernet Groups, RMON Token Ring Extension Groups, RMON2 – The RMON2 Management Information Base, RMON2 Conformance Specifications; ATM Remote Monitoring, A Case Study of Internet Traffic Using RMON.

UNIT 6 **6 Hours**

Broadband Network Management: ATM Networks: Broadband Networks and Services, ATM Technology – Virtual Path-Virtual Circuit, TM Packet Size, Integrated Service, SONET, ATM LAN Emulation, Virtual LAN; ATM Network Management – The ATM Network Reference Model, The Integrated Local Management Interface, The ATM Management Information Base, The Role of SNMP and ILMI in ATM Management, M1 Interface: Management of ATM Network Element, M2 Interface: Management of Private Networks, M3 Interface: Customer Network Management of Public Networks, M4 Interface: Public Network Management, Management of LAN Emulation, ATM Digital Exchange Interface Management.

UNIT 7 **6 Hours**

Broadband Network Management: Broadband Access Networks and Technologies – Broadband Access Networks, roadband Access Technology; HFCT Technology – The Broadband LAN, The Cable Modem, The Cable Modem Termination System, The HFC Plant, The RF Spectrum for Cable Modem; Data Over Cable Reference Architecture; HFC Management – Cable Modem and CMTS Management, HFC Link Management, RF Spectrum Management, DSL Technology; Asymmetric Digital Subscriber Line Technology – Role of the ADSL Access Network in an Overall Network, ADSL Architecture, ADSL Channeling Schemes, ADSL Encoding Schemes; ADSL Management – ADSL Network Management Elements, ADSL Configuration Management, ADSL Fault Management, ADSL Performance Management, SNMP-Based ADSL Line MIB, MIB Integration with Interfaces Groups in MIB-2, ADSL Configuration Profiles.

UNIT 8 **8Hours**

Network Management Applications: Configuration Management- Network Provisioning, Inventory Management, Network Topology, Fault Management- Fault Detection, Fault Location and Isolation Techniques, Performance Management – Performance Metrics, Data Monitoring, Problem

Isolation, Performance Statistics; Event Correlation Techniques – Rule-Based Reasoning, Model-Based Reasoning, Case-Based Reasoning, Codebook correlation Model, State Transition Graph Model, Finite State Machine Model, Security Management – Policies and Procedures, Security Breaches and the Resources Needed to Prevent Them, Firewalls, Cryptography, Authentication and Authorization, Client/Server Authentication Systems, Messages Transfer Security, Protection of Networks from Virus Attacks, Accounting Management, Report Management, Policy-Based Management, Service Level Management.

Text Books:

1. Mani Subramanian: Network Management- Principles and Practice, 2nd Edition, Pearson Education, 2010.

Reference Books:

1. J. Richard Burke: Network management Concepts and Practices: a Hands-On Approach, PHI, 2008.

INFORMATION AND NETWORK SECURITY

Subject Code: 10CS835	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT 1 **6 Hours**

Planning for Security: Introduction; Information Security Policy, Standards, and Practices; The Information Security Blue Print; Contingency plan and a model for contingency plan

UNIT 2 **6 Hours**

Security Technology-1: Introduction; Physical design; Firewalls; Protecting Remote Connections

UNIT 3 **6 Hours**

Security Technology – 2: Introduction; Intrusion Detection Systems (IDS); Honey Pots, Honey Nets, and Padded cell systems; Scanning and Analysis Tools

UNIT 4 **8 Hours**

Cryptography: Introduction; A short History of Cryptography; Principles of Cryptography; Cryptography Tools; Attacks on Cryptosystems.

PART - B

UNIT 5 **8 Hours**

Introduction to Network Security, Authentication Applications: Attacks, services, and Mechanisms; Security Attacks; Security Services; A model for Internetwork Security; Internet Standards and RFCs Kerberos, X.509 Directory Authentication Service.

UNIT 6 **6 Hours**

Electronic Mail Security: Pretty Good Privacy (PGP); S/MIME

UNIT 7 **6 Hours**

IP Security: IP Security Overview; IP Security Architecture; Authentication Header; Encapsulating Security Payload; Combining Security Associations; Key Management.

UNIT 8 **6 Hours**

Web Security: Web security requirements; Secure Socket layer (SSL) and Transport layer Security (TLS); Secure Electronic Transaction (SET)

Text Books:

1. Michael E. Whitman and Herbert J. Mattord: Principles of Information Security, 2nd Edition, Cengage Learning, 2005. (Chapters 5, 6, 7, 8; Exclude the topics not mentioned in the syllabus)
2. William Stallings: Network Security Essentials: Applications and Standards, 3rd Edition, Pearson Education, 2007. (Chapters: 1, 4, 5, 6, 7, 8)

Reference Book:

1. Behrouz A. Forouzan: Cryptography and Network Security, Special Indian Edition, Tata McGraw-Hill, 2007.

MICROCONTROLLER-BASED SYSTEMS

Subject Code: 10CS836	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT 1 **7 Hours**

Introduction, 8051 Assembly Language Programming – 1: Microcontrollers and embedded processors; Overview of the 8051 family 8051 Assembly Language Programming (ALP) -1: Inside the 8051; Introduction to 8051 ALP; Assembling and running an 8051 program; The

PC and ROM space in 8051; Data types, directives, flag bits, PSW register, register banks, and the stack.

UNIT 2 **6 Hours**

ALP – 2 : Jump and loop instructions; Call instructions; Time delay for various 8051 family members; I/O programming; I/O bit manipulation programming. Immediate and register addressing modes; Accessing memory using various addressing modes.

UNIT 3 **7 Hours**

ALP – 3 - Programming in C: Bit addresses for I/O and RAM; Extra 128 bytes of on-chip RAM in 8052.Arithmetic instructions; Signed numbers and arithmetic operations; Logic and compare instructions; rotate instruction and serialization; BCD, ASCII, and other application programs. Programming in C: Data types and time delays; I/O programming; Logic operations; Data conversion programs; Accessing code ROM space; Data serialization.

UNIT 4 **6 Hours**

Pin Description, Timer Programming: Pin description of 8051; Intel Hex file; Programming the 8051 timers; Counter programming; Programming Timers 0 and 1 in C.

PART – B

UNIT 5 **6 Hours**

Serial Port Programming, Interrupt Programming: Basics of serial communications; 8051 connections to RS232; Serial port programming in assembly and in C 8051 interrupts; Programming timer interrupts; Programming external hardware interrupts; Programming the serial communications interrupt; Interrupt priority in 8051 / 8052; Interrupt programming in C.

UNIT 6 **7 Hours**

Interfacing LCD, Keyboard, ADC, DAC and Sensors : LCE interfacing; Keyboard interfacing; Parallel and serial ADC; DAC interfacing; Sensor interfacing and signal conditioning

UNIT 7 **7 Hours**

Interfacing to External Memory, Interfacing with 8255: Memory address decoding; Interfacing 8031 / 8051 with external ROM; 8051 data memory space; Accessing external data memory in C. Interfacing with 8255; Programming 8255 in C.

UNIT 8 **6 Hours**

DS12887 RTC interfacing and Programming, Applications : DS12887 RTC interfacing; DS12887 RTC programming in C; Alarm, SQW, and IRQ features of DS12886 Relays and opto-isolators; Stepper motor interfacing; DC motor interfacing and PWM

Text Books:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay: The 8051 Microcontroller and Embedded Systems using Assembly and C, 2nd Edition, Pearson Education,2008.

Reference Books:

1. Raj Kamal: Microcontrollers Architecture, Programming, Interfacing and System Design, Pearson Education, 2007.
2. Dr. Ramani Kalpathi, Ganesh Raja: Microcontrollers and Applications, 1st Revised Edition, Sanguine - Pearson, 2010.

ADHOC NETWORKS

Sub Code: 10CS841	**IA Marks : 25**
Hrs/Week: 04	**Exam Hours : 03**
Total Hrs: 52	**Exam Marks : 100**

PART – A

UNIT 1 **6 Hours**

Introduction: Ad hoc Networks: Introduction, Issues in Ad hoc wireless networks, Ad hoc wireless internet.

UNIT 2

7 Hours

MAC – 1: MAC Protocols for Ad hoc wireless Networks: Introduction, Issues in designing a MAC protocol for Ad hoc wireless Networks, Design goals of a MAC protocol for Ad hoc wireless Networks, Classification of MAC protocols, Contention based protocols with reservation mechanisms.

UNIT 3 **6 Hours**

MAC – 2: Contention-based MAC protocols with scheduling mechanism, MAC protocols that use directional antennas, Other MAC protocols.

UNIT 4

7 Hours

Routing – 1: Routing protocols for Ad hoc wireless Networks: Introduction, Issues in designing a routing protocol for Ad hoc wireless Networks, Classification of routing protocols, Table drive routing protocol, On-demand routing protocol.

PART- B

UNIT 5 **6 Hours**

Routing – 2: Hybrid routing protocol, Routing protocols with effective flooding mechanisms, Hierarchical routing protocols, Power aware routing protocols

UNIT 6 **7 Hours**

Transport Layer: Transport layer protocols for Ad hoc wireless Networks: Introduction, Issues in designing a transport layer protocol for Ad hoc wireless Networks, Design goals of a transport layer protocol for Ad hoc wireless Networks, Classification of transport layer solutions, TCP over Ad hoc wireless Networks, Other transport layer protocols for Ad hoc wireless Networks.

UNIT 7 **6 Hours**

Security: Security: Security in wireless Ad hoc wireless Networks, Network security requirements, Issues & challenges in security provisioning, Network security attacks, Key management, Secure routing in Ad hoc wireless Networks.

UNIT 8 **7 Hours**

QoS: Quality of service in Ad hoc wireless Networks: Introduction, Issues and challenges in providing QoS in Ad hoc wireless Networks, Classification of QoS solutions, MAC layer solutions, network layer solutions.

Text Books:

1. C. Siva Ram Murthy & B. S. Manoj: Ad hoc Wireless Networks, 2nd Edition, Pearson Education, 2005

Reference Books:

1. Ozan K. Tonguz and Gianguigi Ferrari: Ad hoc Wireless Networks, John Wiley, 2007.
2. Xiuzhen Cheng, Xiao Hung, Ding-Zhu Du: Ad hoc Wireless Networking, Kluwer Academic Publishers, 2004.
3. C.K. Toh: Adhoc Mobile Wireless Networks- Protocols and Systems, Pearson Education, 2002.

SOFTWARE TESTING

Subject Code: 10CS842	**I.A. Marks: 25**
Hours/Week: 4	**Exam Marks: 100**
Total Hours: 52	**Exam Hours: 3**

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, 1314, 15)
2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009.
   (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.

ARM BASED SYSTEM DESIGN

Subject Code: 10CS843	**I.A. Marks: 25**
Hours/Week: 4	**Exam Marks: 100**
Total Hours: 52	**Exam Hours: 3**

PART – A

UNIT 1 **6 Hours**

Introduction: The RISC design philosophy; The ARN design philosophy; Embedded system hardware and software.ARM processor fundamentals: Registers; Current Program Status Register; Pipeline; Exceptions, interrupts and the Vector Table; Core extensions; Architecture revisions; ARM processor families.

UNIT 2 **7 Hours**

ARM Instruction Set and Thumb Instruction Set: ARM instruction set: Data processing instructions; Branch instructions; Load-store instructions; Software interrupt instruction; Program Status Register functions; Loading constants; ARMv5E extensions; Conditional execution.Thumb instruction set: Thumb register usage; ARM –Thumb interworking; Other branch instructions; Data processing instructions; Single-Register Load-Store instructions; Multiple-Register Load-Store instructions; Stack instructions; Software interrupt instruction.

UNIT 3 **6 Hours**

Writing and Optimizing ARM Assembly Code: Writing assembly code; Profiling and cycle counting; Instruction scheduling; Register allocation; Conditional execution; Looping constructs; Bit manipulation; Efficient switches; Handling unaligned data.

UNIT 4 **7 Hours**

Optimized Primitives: Double-precision integer multiplication; Integer normalization and count leading zeros; Division; Square roots; Transcendental functions; Endian reversal and bit operations; Saturated and rounded arithmetic; Random number generation.

PART - B

UNIT 5 **7 Hours**

Exception and Interrupt Handling: Exception handling; Interrupts and interrupt handling schemes

UNIT 6 **7 Hours**

Caches : The memory hierarchy and the cache memory; Cache architecture; Cache policy; Coprocessor 15 and cache; Flusing and cleaning cache memory; Cache lockdown; Caches and software performance.

UNIT 7 **6 Hours**

Memory – 1: Memory Protection Units: Protected regions; Initializing the MPU, cache and write buffer; Demonstration of an MPU system. Memory Management Units: Moving from MPU to an MMU; How virtual memory works; Details of the ARM MMU.

UNIT 8 **6 Hours**

Memory – 2: Page tables; The translation lookaside buffer; Domains and memory access permission; The caches and write buffer; Coprocessor 15 and MMU configuration; The fast context switch extension.

Text Books:

1. Andrew N. Sloss, Dominic Symes, Chris Wright: ARM System Developer‟s Guide – Designing and Optimizing System Software, Elsevier, 2004.

Reference Books:

1. David Seal (Editor): ARM Architecture Reference Manual, 2nd Edition, Addison-Wesley, 2001.
2. Steve Furber: ARM System-on-Chip Architecture, 2nd Edition, Addison-Wesley, 2000.

SERVICES ORIENTED ARCHITECTURE

Subject Code: 10CS844	**I.A. Marks: 25**
Hours/Week: 4	**Exam Marks: 100**
Total Hours: 52	**Exam Hours: 3**

PART – A

UNIT 1 **7 Hours**

Introduction o SOA, Evolution of SOA: Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA;An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures).

UNIT 2 **6 Hours**

Web Services and Primitive SOA : The Web services framework; Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP).

UNIT 3 **6 Hours**

Web Services and Contemporary SOA – 1: Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography

UNIT 4 **7 Hours**

Web Services and Contemporary SOA – 2: Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing

PART – B

UNIT 5 **7 Hours**

Principles of Service – Orientation: Services-orientation and the enterprise; Anatomy of a service-oriented architecture; Common Principles of Service-orientation; How service orientation principles inter-relate; Service-orientation and object-orientation; Native Web service support for service-orientation principles.

UNIT 6 **6 Hours**

Service Layers: Service-orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios

UNIT 7 **7 Hours**

Business Process Design: WS-BPEL language basics; WS-Coordination overview; Service-oriented business process design; WS-addressing language basics; WS-Reliable Messaging language basics

UNIT 8 **6 Hours**

SOA Platforms: SOA platform basics; SOA support in J2EE; SOA support in .NET; Integration considerations

Text Books:

1. Thomas Erl: Service-Oriented Architecture – Concepts, Technology, and Design, Pearson Education, 2005.

Reference Books:

1. Eric Newcomer, Greg Lomow: Understanding SOA with Web Services, Pearson Education, 2005.

Clouds, Grids, and Clusters

Subject Code: 10CS845 **I.A. Marks: 25**
Hours/Week: 4 **Exam Marks: 100**
Total Hours: 52 **Exam Hours: 3**

PART – A

UNIT - 1 **6 Hours**

Introduction: Overview of Cloud Computing, Applications, Intranets and the Cloud, When can cloud Computing be used? Benefits and limitations, Security concerns, Regulatory issues

UNIT - 2 **6 Hours**

Business Case for Cloud, Examples of Cloud Services: Cloud computing services, Help to the business, Deleting the data center. Examples: Google, Microsoft, IBM, Salesforce.com and its uses, Cloud at Thomson Reuters.

UNIT - 3 **7 Hours**

Technology, Cloud Storage, Standards: Cloud Computing Technology: Clients, Security, Network, Services.
Overview of Cloud storage, Some providers of Cloud storage. Standards: Applications, Clients, Infrastructure, Service.

UNIT - 4 **7 Hours**

Other issues: Overview of SaaS (Software as a Service), Driving forces, Company offerings: Google, Microsoft, IBM. Software plus Service: Overview, Mobile device integration Local Clouds, Thin Clients, Migrating to the Cloud: Virtualization, Server solutions, Thin clients, Cloud services for individuals, mid-markets, and enterprises, Migration.

PART - B

UNIT - 5 **7 Hours**

GRID Computing – 1: Introduction: Data Center, The Grid and the Distributed/ High Performance Computing, Cluster Computing and Grid Computing, Metacomputing – the Precursor of Grid Computing, Scientific, Business and e-Governance Grids, Web services and Grid Computing, Business Computing and the Grid – a Potential Win win Situation, e-Governance and the Grid. Technologies and Architectures for Grid Computing: Clustering and Grid Computing, Issues in Data Grids, Key Functional Requirements in Grid Computing, Standards for Grid Computing , Recent Technological Trends in Large Data Grids.OGSA and WSRF: OGSA for Resource Distribution, Stateful Web Services in OGSA, WSRF (Web

Services Resource Framework), Resource Approach to Stateful Services, WSRF Specification.
The Grid and the Database: Issues in Database Integration with the Grid, The Requirements of a Grid enabled database, Storage Request Broker (SRB), How to integrate the Database with the Grid? The Architecture of OGSA-DAI for Offering Grid Database Services

UNIT - 6 **6 Hours**

GRID Computing – 2: World Wide Grid Computing Activites, Organizations and Projects: Standards Organizations, Organizations Developing Grid Computing Tool Kits, Framework and Middleware, Grid Projects and Organizations Building and Using Grid Based Solutions.Web Services and the Service Oriented Architecture (SOA): History and Background, Service Oriented Architecture, How a Web Service Works, SOAP and WSDL, Description, Creating Web Services, Server Side. Globus Toolkit: History of Globus Toolkit, Versions of Globus Toolkit, Applications of GT4 – cases, GT4 – Approaches and Benefits, Infrastructure Management, Monitoring and Discovery, Security, Data, Choreography and Coordination, Main Features of GT4 Functionality – a Summary, GT4 Architecture, GT4 Command Line Programs, GT4 Containers.

UNIT - 7 **7 Hours**

Cluster Computing – 1: Introduction: What is Cluster Computing, Approaches to Parallel Computing, How to Achieve Low Cost Parallel Computing through Clusters, Definition and Architecture of a Cluster, What is the Functionality a Cluster can offer? Categories of Clusters Cluster Middleware: Levels and Layers of Single System Image (SSI), Cluster Middleware Design Objectives, Resource Management and Scheduling, Cluster Programming Environment and Tools. Early Cluster Architectures and High Throughput Computing Clusters: Early Cluster Architectures, High Throughput Computing Clusters, Condor. Setting up and Administering a Cluster: How to set up a Simple Cluster? Design considerations for the Front End of a Cluster, Setting up nodes, Clusters of Clusters or Metaclusters, System Monitoring, Directory Services inside the Clusters & DCE, Global Clocks Sync, Administering heterogeneous Clusters.

UNIT - 8 **6 Hours**

Cluster Computing – 2: Cluster Technology for High Availability: Highly Available Clusters, High Availability Parallel Computing, Mission Critical (or Business Critical or Business Continuity) Applications, Types of Failures and Errors, Cluster Architectures and Configurations for High Availability, Faults and Error Detection, Failure Recovery, Failover / Recovery Clusters. Performance Model and Simulation: Performance Measures and Metrics, Profit Effectiveness of Parallel Computing through Clusters. Process Scheduling, Load Sharing and Load Balancing: Job Management System

(JMS) Resource Management System (RMS), Queues, Hosts, Resources, Jobs and Policies, Policies for Resource Utilization, Scheduling Policies Load Sharing and Load Balancing, Strategies for Load Balancing, Modeling Parameters Case Studies of Cluster Systems: Beowulf, PARAM.

Text Books:

1. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter: Cloud Computing, A Practical Approach, McGraw Fill, 2010.
2. Prabhu: Grid and Cluster Computing, PHI, 2008.

Reference Books:

1. Joshy Joseph, Craig Fellenstein: Grid Computing, Pearson Education, 2007.
2. Internet Resources

MULTI-CORE ARCHITECTURE AND PROGRAMMING

Subject Code: 10CS846	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT 1 **7 Hours**

Introduction

The power and potential of parallelism, Examining sequential and parallel programs, Parallelism using multiple instruction streams, The Goals: Scalability and performance portability, Balancing machine specifics with portability, A look at six parallel computers: Chip multiprocessors, Symmetric multiprocessor architectures, Heterogeneous chip designs, Clusters, Supercomputers, Observations from the six parallel computers.

UNIT 2 **6 Hours**

Reasoning about Performance

Motivation and basic concepts, Sources of performance loss, Parallel structure, Performance trade-offs, Measuring performance, Scalable performance.

UNIT 3 **6 Hours**

Examples of Multi-Core Architectures

Introduction to Intel Architecture, How an Intel Architecture System works, Basic Components of the Intel Core 2 Duo Processor: The CPU, Memory Controller, I/O Controller; Intel Core i7: Architecture, The Intel Core i7 Processor, Intel QuickPath Interconnect, The SCH; Intel Atom Architecture.

Introduction to Texas Instruments‟ Multi-Core Multilayer SoC architecture for communications, infrastructure equipment

UNIT 4 **7 Hours**

Parallel Algorithm Design

Introduction, The Task / Channel model, Foster‟s design methodology, Examples: Boundary value problem, Finding the maximum, The n-Body problem, Adding data input.

PART – B

UNIT 5 **7 Hours**

Parallel Programming – 1 (Using OpenMP)

Designing for threads: Task decomposition, Data decomposition, Data flow decomposition, Implications of different decompositions; Challenges in decomposition, Parallel programming patters, A motivating problem: Error diffusion.

Threading and Parallel Programming Constructs: Synchronization, Critical sections, Deadlocks, Synchronization primitives: Semaphores, Locks, Condition variables; Messages, Flow Control-Based concepts: Fence, Barrier; Implementation-Dependent threading issues.

UNIT 6 **6 Hours**

Parallel Programming – 2 (Using OpenMP)

Introduction, The shared-memory model, Parallel *for* loops, Declaring private variables, Critical sections, Reductions, Performance improvements, More general data parallelism, Functional parallelism.

UNIT 7 **7 Hours**

Solutions to Common Parallel Programming Problems

Too many threads, Data races, deadlocks, and live locks, Heavily contended locks, Non-blocking algorithms, Thread-safe functions and libraries, Memory issues, Cache-related issues, Avoiding pipeline stalls, Data organization for high performance.

UNIT 8 **6 Hours**

Threading in the Processor

Single-Core Processors: Processor architecture fundamentals, Comparing Superscalar and EPIC architectures.

Multi-Core Processors: Hardware-based threading, Hyper-threading technology, Multi-Core processors, Multiple processor interactions, Power consumption, Beyond multi-core architecture.

NOTE: In order to acquire a sound understanding of the subject, it is desirable for the students to work in the laboratory using OpenMP. The

hands-on experience would reinforce the concepts learnt in theory. Problems similar to the ones solved in the Algorithms Laboratory can be solved and issues like speed-up achieved can be analyzed in depth. Several free tools are available from companies like INTEL to facilitate such a study.

Text Books:

1. Calvin Lin, Lawrence Snyder: Principles of Parallel Programming, Pearson Education, 2009.
   (Listed topics only from Chapters 1, 2, 3)
2. Michael J. Quinn: Parallel Programming in C with MPI and OpenMP, Tata McGraw Hill, 2004.
   (Listed topics only from Chapters 3, 17)
3. Shameem Akhter, Jason Roberts: Multi-Core Programming, Increasing Performance through Software Multithreading, Intel Press, 2006.
   (Listed topics only from Chapters 3, 4, 7, 9, 10)
4. Web resources for Example Architectures of INTEL and Texas Instruments:
   http://download.intel.com/design/intarch/papers/321087.pdf ;
   http://focus.ti.com/lit/wp/spry133/spry133.pdf

Reference Books:

1. Introduction to Parallel Computing – Ananth Grama et. al., Pearson Education, 2009.
2. Reinders : Intel Threading Building Blocks, O"reilly – 2005
3. David Culler et. al.: Parallel Computer Architecture: A Hardware/Software Approach, Elsevier, 2006.
4. Richard Gerber, Aart J.C. Bik, Kevin B. Smith, Xinmin Tian: Software Optimization Cookbook, High-Performance Recipes for IA-32 Platforms, 2nd Edition, Intel Press, 2006.