INFORMATION TECHNOLOGY COURSE STRUCTURE
B. Tech. (AUTONOMOUS)
Duration: 4 years (Eight Semesters)

1st SEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Basic Science Course	UBSCH101	CHEMISTRY	3	1	0	4	30	70	0	100
2	Basic Science Course	UBSMH102	MATHEMATICS - I	3	1	0	4	30	70	0	100
3	Engineering Science Course	UESCS103	PROGRAMMING FOR PROBLEM SOLVING	3	1	0	4	30	70	0	100
4	Basic Science Course	ULCCH101	CHEMISTRY LAB	0	0	3	1.5	0	0	100	100
5	Engineering Science Course	ULCCS103	PROGRAMMING FOR PROBLEM SOLVING LAB	0	0	2	1	0	0	100	100
6	Engineering Science Course	ULCME104	ENGINEERING GRAPHICS AND DESIGN LAB	1	0	4	3	0	0	100	100
7	Mandatory Course	INDUCTION TRAINING(21 DAYS)					0				
			Total				**17.5**				**600**

2nd SEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Basic Science Course	UBSPH111	PHYSICS	3	1	0	4	30	70	0	100
2	Basic Science Course	UBSMH202	MATHEMATICS-II	3	1	0	4	30	70	0	100
3	Engineering Science Course	UESEE113	BASIC ELECTRICAL ENGG.	3	0	0	3	30	70	0	100
4	Humanities &Social Sciences	UHSMH205	ENGLISH	2	0	0	2	30	70	0	100
5	Basic Science Course	ULCPH111	PHYSICS LAB	0	0	3	1.5	0	0	100	100
6	Engineerin	ULCEE113	BASIC	0	0	4	2	0	0	10	100

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

	g Science Course		ELECTRICAL ENGG. LAB							0	
7	Engineering Science Course	ULCME114	WORK SHOP/BASIC MANUFACTURING PROCESS LAB	1	0	4	3	0	0	100	100
8	HS	ULCMH204	ENGLISH LAB	0	0	2	1	0	0	100	100
			Total				**20.5**				**800**
9	**Summer Internship programme (4 to 8 weeks) is mandatory as per AICTE rule**										

3rd SEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Core Course	UPCIT301	Object Oriented Programming using JAVA	3	0	0	3	30	70	0	100
2	Core Course	UPCIT302	Data structure using C	3	0	0	3	30	70	0	100
3	Core Course	UPCIT303	Formal Language and Automata Theory	3	1	0	4	30	70	0	100
4	Engg. Science Course	UESIE312	Digital Electronics	3	0	0	3	30	70	0	100
5	Basic Science Course	UBSMH301	Mathematics -III	3	1	0	4	30	70	0	100
6	Humanities Science Course	UHSMH211	Engineering Economics	3	0	0	3	30	70	0	100
7	Lab Course	ULCIT301	JAVA Lab	0	0	3	1.5	0	0	100	100
8	Lab Course	ULCIT302	Data structure Lab	0	0	3	1.5	0	0	100	100
			Total				**23**				**800**

4thSEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Core Course	UPCIT401	Discrete Mathematics	3	0	0	3	30	70	0	100
2	Core Course	UPCIT402	Design & Analysis of Algorithm	3	1	0	4	30	70	0	100
3	Core Course	UPCIT403	Database Management Systems	3	0	0	3	30	70	0	100

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

4	Engg. Science Course	UESIT404	Computer Organization and Architecture	3	0	0	3	30	70	0	100
5	Humanities Science Course	UHSMH212	Organizational Behavior	3	0	0	3	30	70	0	100
6	Lab Course	ULCIT401	Computer Organization Lab	0	0	3	1.5	0	0	100	100
7	Lab Course	ULCIT402	Algorithm Lab	0	0	3	1.5	0	0	100	100
8	Lab Course	ULCIT403	Database Lab	0	0	3	1.5	0	0	100	100
9	Mandatory Course	UMCCE401	Environmental Science	2	0	0	0	30	70	0	100
			Total				**20.5**				**900**
10	**Summer Internship programme (4 to 8 weeks) is mandatory as per AICTE rule**										

5thSEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Core Course	UPCIT501	Operating System	3	0	0	3	30	70	0	100
2	Core Course	UPCIT502	Computer Networks	3	0	0	3	30	70	0	100
3	Core Course	UPCIT503	Compiler Design	3	0	0	3	30	70	0	100
4	Core Course	UPCIT504	Artificial Intelligence	3	0	0	3	30	70	0	100
5	Programme Elective-I	UPEIT505 UPEIT506 UPEIT507	E-Commerce and ERP Adv Java/ Advanced Computer Architecure	3	0	0	3	30	70	0	100
6	Open Elective-I			3	0	0	3	30	70	0	100
7	Lab Course	ULCIT501	Operating System Lab	0	0	3	1.5	0	0	100	100
8	Lab Course	ULCIT502	Computer Networks Lab	0	0	3	1.5	0	0	100	100
9	Lab Course	ULCIT503	Compiler Design Lab	0	0	3	1.5	0	0	100	100
			Total				**22.5**				**900**

6thSEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week	Credit	Maximum Marks

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

INFORMATION TECHNOLOGY COURSE STRUCTURE
B. Tech. (AUTONOMOUS)
Duration: 4 years (Eight Semesters)

				L	T	P		IA	EA	PA	Total
1	Core Course	UPCIT601	Internet & Web Technology	3	0	0	3	30	70	0	100
2	Core Course	UPCIT602	Software Engineering	3	0	0	3	30	70	0	100
3	Programme Elective-II	UPEIT601 UPEIT602 UPEIT603	Data Mining/ Data Analytics/ Computer Vision	3	0	0	3	30	70	0	100
4	Programme Elective-III	UPEIT604 UPEIT605 UPEIT606	Cloud Computing Computer Graphics Soft Computing	3	0	0	3	30	70	0	100
5	Open Elective-II			3	0	0	3	30	70	0	100
6	Lab Course	ULCIT601	IWT Lab	0	0	3	1.5	0	0	100	100
7	Lab Course	ULCIT601	Software Engineering Lab	0	0	3	1.5	0	0	100	100
8	Lab Course	ULCIT601	Simulation Lab	0	0	4	2	0	0	100	100
			Total				**20**				**800**
9	**Summer Internship programme (4 to 8 weeks) is mandatory as per AICTE rule**										

7th SEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Programme Elective-IV	UPEIT701/ UPEIT702/ UPEIT703	Mobile Computing/ Real Time Systems/ Wireless Sensor Network	3	0	0	3	30	70	0	100
2	Programme Elective-V	UPEIT704 UPEIT705 UPEIT706	Software Project Management Information Retrieval Fault Tolerant System/	3	0	0	3	30	70	0	100
3	Open Elective-III			3	0	0	3	30	70	0	100

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

4	Open Elective-IV			3	0	0	3	30	70	0	100
5	Humanities Science Course	UHSMH701	Entrepreneurship Development	3	0	0	3	30	70	0	100
6	Project Course	UPRIT701	Project Stage-1	0	0	6	3	0	0	100	100
7	Seminar	USEIT701	Internship Seminar	0	0	2	1	0	0	100	100
			Total				**19**				**700**

8thSEMESTER

Sl. No.	Subject Type	Subject Code	Subject Name	Teaching Hours/Week			Credit	Maximum Marks			
				L	T	P		IA	EA	PA	Total
1	Programme Elective-VI	UPEIT801 UPEIT802 UPEIT803	Machine Learning/ Embedded Systems / Intrusion Detection System	3	0	0	3	30	70	0	100
2	Open Elective-V			3	0	0	3	30	70	0	100
3	Open Elective-VI			3	0	0	3	30	70	0	100
4	Project Course	UPRIT801	Project Stage-2	0	0	14	7	0	0	100	100
5	CoreCourse	UPCIT801	Comprehensive VivaVoce	0	0	2	1	0	0	100	100
			Total				**17**				**500**

List of Open Elective Courses offered by IT Department

1. **Data Structure**
2. **Object Oriented Programming using C++**
3. **Java Programming**
4. **Data Base Engineering**
5. **Computer Organization**
6. **Data Mining**

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

INFORMATION TECHNOLOGY COURSE STRUCTURE
B. Tech. (AUTONOMOUS)
Duration: 4 years (Eight Semesters)

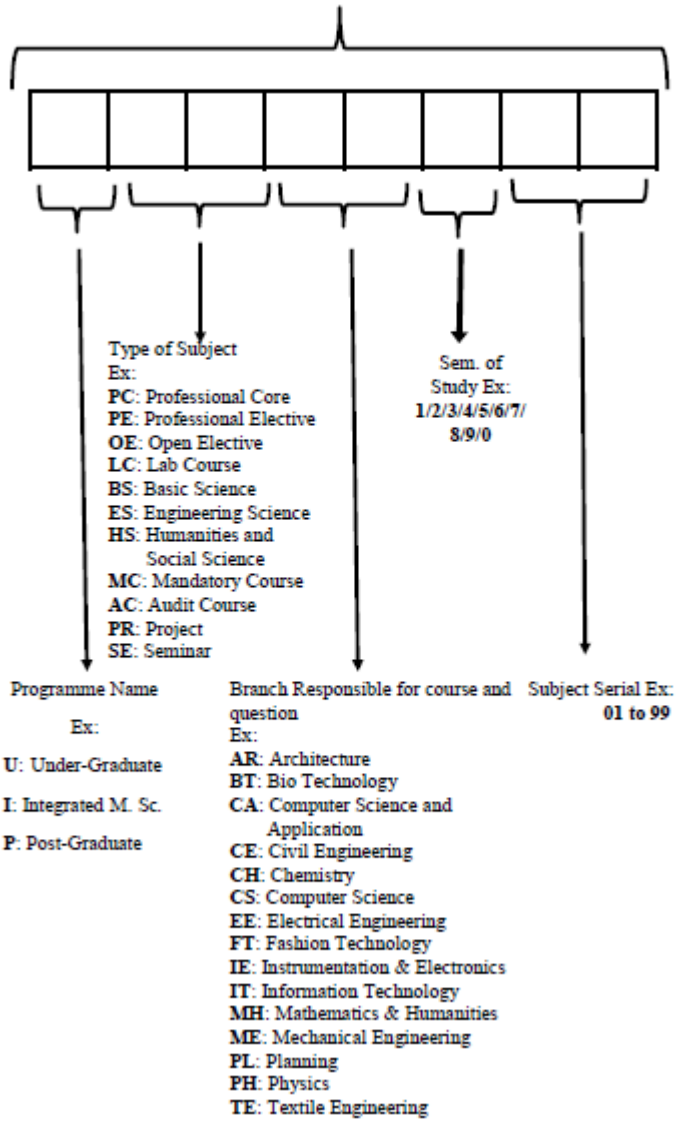
Semester Wise Credits Break Up								
Subject Type **Semester**	**Professional Core**	**Basic Science**	**Engineering Science**	**Humanities**	**Program Elective**	**Open Elective**	**Project /Seminar**	**Total**
1st		9.5	8					**17.5**
2nd		9.5	8	3				**20.5**
3rd	13	4	3	3				**23**
4th	14.5		3	3				**20.5**
5th	16.5				3	3		**22.5**
6th	11				6	3		**20**
7th				3	6	6	4	**19**
8th					3	6	8	**17**
Total	**55**	**23**	**22**	**12**	**18**	**18**	**12**	**160**

Abbreviations Used: **L = Lectures, P = Practical or Laboratory, T = Tutorial**

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

INFORMATION TECHNOLOGY COURSE STRUCTURE
B. Tech. (AUTONOMOUS)
Duration: 4 years (Eight Semesters)

SUBJECT CODE FORMATION



Abbreviations Used: **L = Lectures, P = Practical or Laboratory, T = Tutorial**

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

THIRD SEMESTER

OBJECT ORIENTED PROGRAMMING USING JAVA

Prerequisites	Problem-solving approach Programming language, Data Structure, Database Engineering
Course Outcomes	• Understand the use of OOPs concepts • Able to solve real world problems using OOP techniques. • Understand the use of abstraction. • Understand the use of Packages and Interface in java. • Develop and understand exception handling, multithreaded applications with synchronization. • Understand the use of Collection Framework. • Design GUI based applications and applets for web applications.

Module I: 10 hrs

Object Oriented Programming Concepts:- Objects and Classes , Abstraction , Encapsulation, Inheritance,Polymorphism, **OOP in Java** -Characteristics of Java ,The Java Environment, Java Source File,Structure,Compilation, **Fundamental Programming Structures in Java** – Defining classes in Java , constructors, methods ,access specifiers, static members ,Comments, Data Types, Variables, Operators, Control Flow, Method overloading, Arrays , Packages, Wrapper class, String class, StringBuffer class, StringTokenizer class, **Exceptions**-exception hierarchy , throwing and catching exceptions , built-in exceptions, creating own exceptions

Module II: 8 hrs

Inheritance – Super classes and sub classes ,Protected members , constructors in sub classes and super class, Method overriding, Dynamic method dispatch, Abstract classes and methods, final methods and classes ,**Interfaces**– defining an interface, implementing interface, differences between classes and interfaces ,extending interfaces , **Object class**-object cloning, **Inner classes**, **Input / Output Basics** – Streams , Byte streams and Character streams, Reading and Writing Files

Module III: 8 hrs

Multi-threaded programming – thread properties ,thread creation, thread life cycle ,interrupting threads , thread synchronization, **Generic Programming**- Motivation for generic programming , generic classes ,generic methods , generic code and virtual machine , inheritance and generics ,reflection and generics, **Collections:-**ArrayList, LinkedList, HashSet, TreeSet, Map, Stack

Module IV: 10 hrs

Graphics programming– Applet class, AWT event hierarchy, Containers:-Lightweight and heavyweight container ,Components – working with 2D shapes, Color class, Font class, and Image class, **Basics of event handling** – event handlers, different types of events and listener interfaces, Adapter classes, Introduction to Swing – Model-View Controller, Controller design pattern ,different components in swing, layout management,**JavaFX**

TEXT BOOK:

1. Java 2: The complete reference by Herbert Schildt, 9th Edition, McGraw Hill Education, 2014.
2. Core Java for Beginners by Rashmi Kant Das, 3rd Edition, VikasPublication,2013

REFERENCES:

1. Programming With Java:A Primer By Balagurusamy, 3rd Edition, TMH, 2007
2. Core Java: Fundamentals Volume –I by Cay S. Horstmann, Gary Cornel, 9th Edition, Prentice Hall, 2013.
3. The Java Language Specification by James Gosling, Bill Joy, Guy L. Steele Jr., Gilad Bracha, Alex Buckley, Java SE 8 Edition Addison-Wesley Professional, 2014.
4. Head First Java: A Brain-Friendly Guide By Kathy Sierra, Bert Bates, 2nd Edition, O'Reilly, 2005

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

DATA STRUCTURE USING C

Prerequisites	C Programming, Basic Mathematics
Course Outcomes	• Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms • Compare and contrast the benefits of dynamic and static data structures implementations • Design and implement an appropriate hashing function for an application • Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.

Module – I (12hrs)

Introduction to data structures, Arrays and Row/Column major representation of Arrays, Sparse matrix, Stack: operation on stack, applications of stack. Queues: representation of queues, types of queues and application, Linked lists: Single linked lists, linked list representation of stacks and Queues, Operations on polynomials, double linked list, circular list.

Module – II (12hrs)

Trees: Binary tree, Binary search tree, AVL Tree,Threaded binary tree, General tree, Complete Binary Tree representation, B-tree, Binary Tree traversal methods, Manipulation of Expression through Binary tree, Operations on Binary Search tree and AVL tree

Graphs: Graph terminology, Representation of graphs, path matrix, BFS (breadth first search), DFS (depth first search), topological sorting

Module –III (12hrs)

Sorting and Searching techniques: Bubble sort, selection sort, Insertion sort, Quick sort, merge sort, Heap sort, Radix sort, Shell sort, Linear and binary search.

Hashing: Hashing techniques, Hash function, Address calculation techniques- common hashing functions Collision resolution, Linear probing, quadratic probing, Double hashing, Rehashing, Memory allocation, garbage collection.

Text Books:

1. An introduction to data structures with applications by J. Tremblay and P. G. Sorenson, 2nd edition, McGraw Hill Education
2. Data Structures & Algorithms by GAV Pai, McGraw Hill, 2008

Reference Books:

1. Data Structure using C by Tanenbaum, Pearson Education, 2009
2. Data Structure- A Pseudo code approach with C by Gilberg and Forouzan, 2nd edition Cengage Learning
3. Fundamentals of Data Structure in C by Horowitz, Sahani& Freed, 2nd edition, Universities Press, 2008.
4. Data Structures with C by Lipschutz (Schaum's Outline Series), McGraw Hill Education, 2010
5. Introduction to Data Structures in C by Ashok N. Kamthane, Pearson, 2009

FORMAL LANGUAGE AND AUTOMATA THEORY

Prerequisites	Computer Programming Language, Data structures and Algorithms.
Course Outcomes	• Understand Formal Languages and its applications. • Understanding the Context free languages and grammars, and also Normalising CFG. • Understanding the minimization of deterministic and nondeterministic finite automata. • Understand basic properties of Turing machines and computing with Turing machines • Understand the basic concepts of Complexity theory and Limits of Computation.

Module –I (9 hours)

Fundamentals: Strings, Alphabet, Languages, Operations on strings, Finite state machine, definitions, finite automaton model, acceptance of strings, deterministic finite automaton and non deterministic finite automaton, transition diagrams and transition table. Language recognizers: NFA with € transitions - Significance, acceptance of languages. Conversions and Equivalence: Equivalence between NFA with and without € transitions, NFA to DFA conversion, Minimisation of Finite Automata, equivalence between two FSM's, Design of DFA, Finite Automata with output- Moore and Melay machines.

Module II: (9 hours)

Regular Languages : Regular sets, operators in regular expressions, identity rules, Building finite Automata from regular expressions, Arden's theorem, Building Regular expression from Finite Automata, Pumping lemma for regular languages, Closure properties of regular sets, CYK Algorithm.

Context Free Grammars: Context free grammar, Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, derivation and Parse trees, and sentential forms, Right most and leftmost derivation of strings, Ambiguity, Elimination of Ambiguity, and Simplification of a CFG, Chomskey and Greibach Normal forms, Closure and Decision properties of CFL, Pumping lemma for CFL.

Module III: (9 hours)

Push down Automata: Push down automata, definition, and model, Components, Moves of a PDA, ID of a PDA, Design of a PDA, PDA to CFG and CFG to PDA conversion, Equivalence of CFL and PDA, Introduction to DCFL and DPDA.

Turing Machine : Turing Machine, definition, model, Components, Moves of TM,ID of TM, Design of a TM ,Computable functions, Recursively enumerable languages. Several of Turing Machine's model, Church's hypothesis, counter machine, types of Turing machines, Universal Turing Machine and Undecidable problems, Undecidability of post correspondence Problem. Linear bounded automata and context sensitive language, Chomsky's Hierarchy of Languages.

Module IV: (9 hours)

Primitive Recursive function: Recursive functions, Cantor and Godel numbering, Ackermann's function, Excursiveness of Ackermann and Turing computable functions. Church Turing Hypothesis, Recursive and Recursively enumerable sets, NP completeness: Definition of P and NP problems, NP complete and NP hard problems.

TEXT BOOKS:

1. Introduction to Automata Theory Languages and Computation by Hop croft and Ullman, 3rd Edition, Pearson Education, 2009
2. Introduction to Theory of Computation by Michael Sipser, 3rd Edition, CENGAGE Learning

REFERENCES BOOKS:

1. Introduction to Formal languages, Automata Theory and Computation by Kamala Krithivasan and Rama R, Pearson Education, 2009.
2. Introduction to Computer Theory, Daniel I.A. Cohen, 2nd Edition, Wiley India, 2008.
3. Theory of Computation by V.Kulkarni, Oxford University Press, 2013.
4. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 3rd Edition, PHI, 2007.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

JAVA LAB

Prerequisites	Problem-solving approach Programming language, Data Structure
Course Outcomes	• Understand object, class, encapsulation concept • Use constructor, this keyword, method overloading, Arrays • Use package, wrapper class, String class, StringBuffer class, StringTokenizer class • Understand the use of abstraction. • Understand the use of Packages and Interface in java. • Develop and understand exception handling, multithreaded applications with synchronization. • Understand the use of Collection Framework. • Design GUI based applications and applets for web applications.

List of experiments to be done(Any 10)

1. Simple Java programs using control structures, class, object, access specifier, static keyword.
2. Programs using constructor, this keyword, method overloading, Arrays
3. Programs using package, wrapper class, String class, StringBuffer class, StringTokenizer class
4. Programs using Exception handling, Inheritance concept, final keyword
5. Programs using Abstract class, Interface, Object cloning, Inner class
6. Programs on reading and writing on files
7. Programs on Multithreading
8. Programs on ArrayList, LinkedList, Set, Map, Stack
9. Programs on 2D graphics using Applet and AWT
10. Programs on event handling and Layout manager
11. Programs on Swing
12. Programs on Generics

DATA STRUCTURE LAB

Prerequisites	C Programming, Basics of Mathematics
Course Outcomes	• Understand elementary data structures such as stacks, queues, linked lists, trees and graphs. • Analyze the appropriate data structure for given problem. • Apply different data structures to represent real world problems • Implement different sorting and searching algorithms.

Experiment No.1

(a) Write a C program that implement Bubble Sort method to sort a given list of integers in ascending order.
(b) Write a C program that implement merge Sort method to sort a given list of integers in ascending order.
(c) Write a C program that implement Quick Sort method to sort a given list of integers in ascending order

Experiment No.2

(a) Write C program that implement the Linear search operation for a Key value in a given list of integers
(b) Write C program that implement the Binary search operation for a Key value in a given list of integers

Experiment No. 3

(a) Write a C program to create a stack using an array and perform (i) push operation (ii) pop operation
(b) Write a C program that uses Stack operations to perform the following: i) Converting infix expression into postfix expression ii) Evaluating the postfix expression

Experiment No. 4

(a) Write a C program to create a queue and perform i) Rear ii) front iii) Traversal
(b) Write a C program to create a circular queue and perform i) insertion ii) deletion iii) Traversal

Experiment No. 5

Write a C program to create a Deque and perform i) insertion ii) deletion iii) Traversal

Experiment No. 6

Write a C program that uses functions to perform the following operations on Single linked list: i) Creation ii) Insertion iii) Deletion iv) Traversal

Experiment No. 7

Write a C program that uses functions to perform the following operations on Double linked list: i) Creation ii) Insertion iii) Deletion

Experiment No. 8

Write a C program that uses functions to perform the following operations on Binary Tree: i) Creation ii) Insertion iii) Deletion

Experiment No.9

Write a C program to implement the depth-first search algorithm.

Experiment No.10

Write a C program to implement the breadth-first search algorithm.

Engineering Economics (3-0-0)

Prerequisites:

1. Mathematics.
2. Basic Economics.

Module 1: (10 Hours)

Engineering Economics: Nature, Scope, Basic problems of an economy, Micro Economics and Macro Economics.

Demand: Meaning of demand, Demand function, Law of Demand and its exceptions, Determinants of demand, Demand Estimation and Forecasting, Elasticity of demand & its measurement (Simple numerical problems to be solved), Supply-Meaning of supply, Law of supply and its exception, Determinants of supply, Elasticity of supply, Determination of market equilibrium (Simple numerical problems to be solved).

Production: Production function, Laws of returns: Law of variable proportion, Law of returns to scale.

Module 2: (10 Hours)

Cost and revenue concepts, Basic understanding of different market structures, Determination of equilibrium price under perfect competition (Simple numerical problems to be solved), Break Even Analysis-linear approach (Simple numerical problems to be solved).

Banking: Commercial bank, Functions of commercial bank, Central bank, Functions of Central Bank.

Inflation: Meaning of inflation, types, causes, measures to control inflation.

National Income: Definition, Concepts of national income, Method of measuring national income.

Module 3: (10 Hours)

Time value of money: Interest - Simple and compound, nominal and effective rate of interest, Cash flow diagrams, Principles of economic equivalence.

Evaluation of engineering projects: Present worth method, Future worth method, Annual worth method, Internal rate of return method, Cost benefit analysis for public projects.

Depreciation: Depreciation of capital assert, causes of depreciation, Methods of calculating depreciation (Straight line method, Declining balance method), After tax comparison of project.

Text Books:

1. Riggs, Bedworth and Randhwa," Engineering Economics", McGraw Hill Education India.
2. Deviga Vengedasalam," Principles of Economics", Oxford University Press.
3. William G. Sullivan, Elin M. Wicks, C. Patric Koelling," Engineering Economy", Pearson.
4. R. Paneer Selvam," Engineering Economics", PHI.
5. S. P. Gupta," Macro Economics", TMH.
6. S. B. Gupta," Monetary Economics", Sultan Chand and Co.

Mathematics-III (3-1-0)

Prerequisites:

1. Mathematics-I
2. Mathematics-II

Course Outcomes

On successful completion of this course, the students will be able to:

1. Have a fundamental knowledge of the concepts of probability theory.
2. Do correlation and regression and fitting of different types of curves.
3. Apply sampling theory and theory of estimation in various engineering problems and do various

tests of hypothesis and significance.

4. Use calculators and tables to perform simple statistical analyses for small samples and use popular statistics packages, such as SAS, SPSS, S-Plus, R or MATLAB to perform simple and sophisticated analyses for large samples.

Module 1: (10 Hours)

Probability: Introduction, Probability of an event, additive rule & multiplication rule, conditional probability, Bayes' rule, random variable, discrete and continuous probability distribution, Joint probability distribution, Mathematical expectations, Variance and Co- variance of random variables, Mean and Co- variance of linear combination of random variables, Chebyshev theorem.

Module 2: (10 Hours)

Discrete Probability Distribution: Binomial & Multinomial, Hyper- geo- metric, Geometric, Poisson distribution.

Continuous Probability Distribution: Uniform, Normal, Exponential Distribution, Weibull's Distribution, Chi-square Distribution, Sampling Distribution: Sampling Distribution of S2, t Distribution, F Distribution.

Module 3: (10 Hours)

Estimation of parameter: methods of estimation, Estimating the mean of a single sample, Standard error, Prediction interval, Tolerance limits, Estimating the difference between means of two samples, estimating proportion and variance of single sample, Estimating the difference between two proportions and variances of two samples, maximum likelihood estimation.

Module 4: (10 Hours)

Testing of hypothesis: one and two tailed test, test on a single mean when variance is known & variance is unknown. Test on two means, test on single mean and two mean populations. One and two sample test for variance. $\chi 2$ test for goodness of fit and test for independence.

Introduction to linear regression: Simple regression models, method of least squares, Properties of least square estimators, Inferences concerning the regression coefficients, Coefficients of determination and its application.

Statistical quality control (Simple Idea only)

Text Books:

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers & Keying Ye," Probability & Statistics for Engineers & Scientists", Eighth Edition, 2007, Pearson Education Inc., New Delhi.
2. Jay L. Devore," Probability and Statistics for Engineering and Sciences", Seventh Edition, Thomson/CENGAGE Learning India Pvt. Ltd.

Reference Books:

1. William Mendenhall, Robert J. Beaver & Barbara M. Beaver," Introduction to Probability and Statistics", 13th Edition, 2009, CENGAGE Learning India Pvt. Ltd., New Delhi.
2. T. Veerarajan," Probability, Statistics and Random Processes", Tata McGraw Hill
3. Ronald Deep," Probability and Statistics", Academic Press

FOURTH SEMESTER

COMPUTER ORGANIZATION AND ARCHITECTURE

Prerequisites	Computer fundamentals, Number Systems, Digital Electronics
Course Outcomes	• Understand the theory and architecture of central processing unit. • Analyze design issues in terms of speed, technology, cost and performance. • Demonstrate how to perform the different arithmetic operations on integers and floating-point numbers using two's complement and IEEE floating point representation. • Exemplify in a better way the memory system and system organization.

Module – I (12 Hrs)

Basic Structure of Computers : Computer Architecture vs Computer Organization, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance

Machine Instructions and Programs : Memory Locations and Addresses, Big-Endian and Little-Endian Assignments, Memory Operations, Instruction Set: Instruction Format, Instruction Types(RISC and CISC architecture), Instruction Execution and Straight-Line Sequencing, Branching, Addressing Modes, Basic Input/Output Operations, Subroutines, Additional Instructions.

Module – II (12 Hrs)

Arithmetic :Fixed point and Floating point representation, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication- Booth's algorithm , Fast Multiplication, Integer Division

Basic Processing Unit : Fundamental Concepts, Execution of a Complete Instruction, Hardwired Control, Microprogrammed Control.

Module – III (12 Hrs)

Memory System : Basic Concepts, Memory Hierarchy, Cache Memory, Cache Memory Mapping Functions, Replacement Algorithms

Multiple-Bus Organization: IO bus - SCSI bus, ISA bus, Bus Control – Synchronous bus, Asynchronous bus, Bus Arbitration

Text Books:

1. Computer Architecture and Organization : John P.Hayes, McGraw Hill, 3rd Edition, 2012
2. Computer Architecture and Organization: William Stallings, Pearson Education 10th Edition, 2017
3. Computer Organization: Carl Hamacher, ZvonkoVranesic, SafwatZaky, McGraw Hill, 5th Edition, 2002

Reference Books:

1. Computer System Architecture: Morris M. Mano, PHI, 3rd edition, 2007
2. Computer Architecture and Organization: Design Principles and Applications By Govindarajalu, 8th Reprint, TMH

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

DESIGN AND ANALYSIS OF ALGORITHM

Prerequisites	Mathematics, Basic program design concepts (e.g., pseudocode), Data structure
Course Outcomes	• Analyze the performance of algorithms. • Choose appropriate algorithm design techniques for solving problems. • Understand how the choice of data structures and the algorithm design methods impact the performance of programs. • Clear up troubles the usage of set of rules design methods including the divide and conquer, dynamic programming, backtracking and greedy approach • Find approximate polynomial solution for NP category problem

Module- I (9 Hours)

Introduction to design and analysis of algorithms, Asymptotic analysis of algorithms, Growth of Functions, Recurrences, solution of recurrences by substitution, Recursion Tree and Master methods

Design & Analysis of Divide and conquer algorithms:- Merge sort, Quick sort , Strassen's matrix multiplication

Module – II (9 Hours)

Greedy Algorithms - Elements of Greedy strategy, Activity- selection Problem, Fractional knapsack problem, Huffman codes

Dynamic programming algorithms: - Elements of dynamic programming, The principle of optimality, Matrix-chain multiplication, Longest common subsequence, Assembly-line scheduling, String matching :-Naive String matching algorithm, Rabin-Karp algorithm

Module – III(10 Hours)

Data structure for disjoint sets:- Disjoint set operations, Linked list representation, Disjoint set forests

Graph Algorithms: -Breadth First Search and Depth-First Search, Minimum Spanning Trees, Kruskal and Prim's algorithms, Single Source Shortest Path (Bellman-ford and Dijkstra's algorithms), All Pairs Shortest Paths (Floyd – Warshall Algorithm).

NP Completeness:-Polynomial time solving, Polynomial time verification, NP - Completeness and reducibility, NP Complete problems (without proof):-Circuit Satisfiability problem, Hamiltonian cycle problem, Travelling Salesman Problem, Vertex Cover Problem

Module – IV (08 Hours)

Back tracking algorithm:-Knapsack problem, N-queen problem, Graph Coloring
Branch and Bound algorithm: - 0/1 knapsack algorithm, 15-puzzle
Polynomials and Fast Fourier Transform (FFT)

Text Book:

1. Introduction to algorithms by T.H. Cormen, C.E. Leiserson, R.L. Rivest, C.Stein, 3rd Edition, MIT Press, 2009.
2. Computer Algorithms by Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, 2nd Edition, Silicon Press, 2008

Reference Book:

1. Fundamentals of algorithms by Gilles Brassard, Paul Brately:, Pearson India , 2015
2. Algorithm Design by Goodrich, Tamassia, Wiley India
3. The Algorithm Design Manual, Steven S. Skiena, Second Edition, Springer

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

DATABASE MANAGEMENT SYSTEM

Prerequisites	Elementary set theory, concepts of relations and functions,, Data Structure, Algorithm, Programming Languages
Course Outcomes	• Explain the underlying concepts of Data Base technology. • Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL. • **Understand** the concepts of transaction processing and represent the issues &technology relate to concurrency and recovery in multi user Environment. • **F**amiliar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing.

Module I: (9 Hours)

Introduction to database Systems, advantages of database system over traditional file system, Basic concepts & Definitions, Database users, Database Language, Database System Architecture, Schemas, Sub Schemas, & Instances, database constraints, 3-level database architecture, Data Abstraction, Data Independence, Mappings, Structure, Components & functions of DBMS, Data models.

Module II (9 Hours)

Entity relationship model, Components of ER model, Mapping E-R model to Relational schema, Network and Object Oriented Data models, Storage Strategies: Detailed Storage Architecture, Storing Data, Magnetic Disk, RAID, Other Disks, Magnetic Tape, Storage Access, File & Record Organization, and File Organizations & Indexes.

Module III: (9 Hours)

Relational Algebra, Tuple & Domain Relational Calculus, Relational Query Languages: SQL and QBE. Database Design: Database development life cycle (DDLC), Automated design tools, Functional dependency and Decomposition, Join strategies, Dependency Preservation & lossless Design, Normalization, Normal forms:1NF, 2NF,3NF, and BCNF, Multi-valued Dependencies, 4NF&5NF. Query processing and optimization: Evaluation of Relational Algebra Expressions, Query optimization, Query cost estimation.

Module IV: (9 Hours)

Transaction processing and concurrency control: Transaction concepts, properties of transaction, concurrency control, locking and Timestamp methods for concurrency control schemes. Database Recovery System, Types of Data Base failure, Types of Database Recovery, Recovery techniques, Fundamental concepts on Object-Oriented Database, Data warehousing & Data Mining and Big data and No SQL

Text Book:

1. Database System Concepts by Sudarshan&Korth, 6th edition, McGraw-Hill Education, 2011.
2. Fundamentals of Database System by Elmasari&Navathe, Pearson Education, 2008.

References Books:

1. Database Management Systems by R. Ramakrishnan, 3rd edition, McGraw-Hill Education, 2003.
2. Database Management Systems By R.Panneerselvam , 3rd Edition, PHI, 2018
3. Introduction to Database Management Systems By AtulKahate, PEARSON

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

COMPUTER ORGANIZATION LAB

Prerequisites	Basic Knowledge of Computer
Course Outcomes	• Understand how to implement memory chips, boards, modules and caches. • Understand the basics of hardwired and micro-programmed control of the CPU. • Learn about various I/O devices and the I/O interface.

Experiment 1 - To recognize various components of PC

Experiment 2 - Dismantling and assembling a PC.

Experiment 3 - Study of Motherboard

Experiment 4 - Study of SMPS

Experiment 5 - Study of Printer

Experiment 6 - Study of Microprocessor

Experiment 7 - Design and verify Half-Adder & Full-Adder using VHDL code.

Experiment 8 - Design and verify multiplexer & de-multiplexer using VHDL code.

Experiment 9 - Simulation of ALU using VHDL code.

Experiment 10 - Simulation of CPU using VHDL code.

ALGORITHM LAB

Prerequisites	Mathematics, Basic program design concepts (e.g., pseudocode), Data structure
Course Outcomes	• Analyze the performance of algorithms. • Choose appropriate algorithm design techniques for solving problems. • Use design methods including the divide and conquer in problem solving • Use dynamic programming approach in problem solving • Use greedy approach in problem solving

Implementation and Analysis of (Any 10)

1. Linear Search and Binary Search Algorithm
2. Quick Sort and Merge Sort Algorithm
3. Heap Sort Algorithm
4. Matrix Chain Multiplication Algorithm
5. Longest Common Subsequence Algorithm
6. Fractional Knapsack Algorithm
7. Huffman Code Algorithm
8. Breadth First Search and Depth First Search Algorithm
9. Kruskal Algorithm and Prim's Algorithm
10. Bellman Ford Algorithm/Floyd-Warshall Algorithm
11. Dijkstra's Algorithm
12. Rabin-Karp pattern matching algorithm

DATABASE LAB

Prerequisites	Elementary set theory, concepts of relations and functions,, Data Structure, Algorithm, Programming Languages
Course Outcomes	• Design and implement a database schema for given problem domain. • Implement a query database using DDL/DML command. • Programming on triggers, packages functions. • Design the programs on JDBC &ODBC using VB/ VC++.

1. Use of SQL syntax: insertion, deletion, join, updation using SQL.
2. Programs on join statements and SQL queries including where clause.
3. Programs on procedures and functions.
4. Programs on database triggers
5. Programs on packages.
6. Programs on data recovery using check point technique.
7. Concurrency control problem using lock operations.
8. Programs on ODBC using either VB or VC++.
9. Programs on JDBC. (1 class)
10. Programs on embedded SQL using C / C++ as host language

Organizational Behaviour (3-0-0)

Prerequisites:

1. English.

Module 1: (10 Hours)

The study of Organizational Behaviour: Definition, Meaning, Why study OB; Learning - Principles of learning and learning theories; Personality- Meaning, Determinants, Types, Personality and OB; Perception- Perceptual Process, perceptual errors, Importance of perception in organizations; Motivation- Nature and Importance, Theories of motivation (Herzberg, Maslow, McGregor).

Module 2: (10 Hours)

Group level: Groups in Organizations -Nature, Types, Reasons behind forming groups, Determinants, factors contributing to Group Cohesiveness, Group Decision Making- Process, advantages and disadvantages; Team- Effective Team Building; Types of Leadership- Effective Leadership, Styles of leadership, Leadership Theories-Trait Theory and Contingency Theory, Leadership and Followership; Conflict- Healthy Vs Unhealthy conflict, Conflict Resolution Techniques.

Module 3: (10 Hours)

Structural level: Organizational Culture: culture and organizational effective- ness; Organizational Change: Types of change, Reasons to change, Resistance to change and to manage resistance. Introduction to organizational development.

Text Books:

1. Stephens P. Robbins, Organizational Behaviour, PHI.
2. K. Aswatthappa, Organizational Behaviour, HPH.

Reference Books:

1. Kavita Singh, Organizational Behaviour, Pearson.
2. D. K. Bhattacharya, Organizational Behaviour, OUP.
3. Pradeep Khandelwal, Organizational Behaviour, TMH.
4. Keith Davis, Organizational Behaviour, McGraw Hill.
5. Nelson Quick, ORGB, Cengage Learning.

FIFTH SEMESTER

OPERATING SYSTEM

Prerequisites	Basics of Computer, Programming Languages, Computer Organisation, Data Structure and Algorithms
Course Outcomes	• Understand role of Operating System in terms of process, memory, file and I/O management. • Apply and analyse the concept of a process, thread, mutual exclusion and deadlock. • Evaluate performance of process scheduling algorithms and IPC. • Apply and analyse the concepts of memory management techniques. Evaluate the performance of memory allocation and replacement techniques. • Apply and analyze different techniques of file and I/O management.

MODULE-I 12 Hours

INTRODUCTION TO OPERATING SYSTEM: Basics of Computer System Architecture and Organization, Simple Batch Systems, Multiprogramming and Time Sharing systems. Personal Computer Systems, Parallel Systems, Distributed Systems and Real time Systems, Operating System Services, Operating System Operations, System calls, Operating System Structures

PROCESS MANAGEMENT: Process Concept, Process Scheduling, Operation on Processes, Interprocess communication, Threads, Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling Basic concepts, scheduling criteria, scheduling algorithms, Thread Scheduling.

MODULE-II 12 Hours

PROCESS SYNCHRONIZATION: The Critical section problem, Peterson's solution, Synchronization hardware, Semaphores, Classical problems of synchronization, Monitors.

DEADLOCKS: System model, Deadlock Characterization Methods for Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, recovery from Deadlock.

MEMORY MANAGEMENT: Memory Management strategies, Logical versus Physical Address space, swapping, contiguous Allocation, Paging, Segmentation.

Virtual Memory: Background, Demand paging, performance of Demand paging, Page Replacement, Page Replacement Algorithms, Allocation of frames, Thrashing, Demand Segmentation.

MODULE-III 11 Hours

STORAGE MANAGEMENT: File System Concept, Access Methods, File System Structure, File System Structure, File System Implementation, Directory implementation, Efficiency and Performance, Recovery, Overview of Mass Storage Structure, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, I/O System Overview, I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O Request to Hardware Operation.

TEXT BOOK:

1. Operating System Concepts – Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 8th edition, Wiley-India, 2009.
2. Operating Systems: Internals And Design Principles- William Stallings, 6th Edition, Pearson, 2009

REFERENCE BOOK:

1. Principles of Operating Systems-Naresh Chauhan, Oxford University Press, 2014
2. Modern Operating Systems – Andrew S. Tanenbaum, 3rd Edition, PHI
3. Operating Systems: A Spiral Approach – Elmasri, Carrick, Levine, TMH Edition
4. Operating Systems – H.M. Deitel, P. J. Deitel, D. R. Choffnes, 3rd Edition, Pearson

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

COMPUTER NETWORKS

Prerequisites	Digital Electronics, Computer Organization
Course Outcomes	• Describe the functions of each layer in OSI and TCP/IP model. • Explain the functions of Application layer and Presentation layer paradigms and Protocols. • Describe the Session layer design issues and Transport layer services. Classify the routing protocols and analyze how to assign the IP addresses for the given network. • Describe the functions of data link layer and explain the protocols. • Explain the types of transmission media with real time applications

Module -I (12Hrs)

Overview of Data Communication Networks, Protocols and standards, OSI Reference model,TCP/IP Protocol. Physical Layer: Analog Signals, Digital Signals, Data Rate Limits, Transmission Impairment, Data rate limit, Digital Transmission: Digital-to-Digital conversion, Analog-to-Digital conversion, Transmission modes, Analog Transmission: Digital-to-Analog conversion, Analog-to-Analog conversion, Multiplexing: Frequency Division Multiplexing (FDM), Wave Division Multiplexing(WDM), Time Division Multiplexing (TDM), Transmission Media: Guided Media (Twisted-Pair Cable, Coaxial Cable and Fiber-Optic Cable) and unguided media (wireless), Switching: Circuit Switched Network, Datagram Network, Virtual-Circuit Network , Telephone Network, Dial-up Modems and Digital Subscriber Lines.

Module-II (9Hrs)

Error Detection and correction: Types of Errors, Error Detection mechanism (Linear codes, CRC, Checksum), Error Correction mechanism: Hamming Encoding. Data Link Control and Protocols: Flow and Error Control, Stop-and-Wait ARQ. Go-Back-N ARQ, Selective Repeat ARQ, HDLC and Point-to-Point Protocol, Multiple Access: Random Access (ALOHA, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Polling, Reservation, Token Passing), Channelization (FDMA, TDMA, CDMA).Wired LANs (Ethernet): Traditional Ethernet, Fast Ethernet, Gigabit Ethernet.

Module-III (9Hrs)

Wireless LANs: IEEE 802.11 and Bluetooth. Connecting Devices: Passive Hub, Repeater, Active Hub, Bridge, Two layers Switch, Router, Three layers Switch, Gateway. Virtual Circuit Networks: Frame Relay, Architecture & layers, ATM: Design goals, Architecture &layers. Network Layer: IPV4 addresses, IPV6 addresses, Internet Protocol: Internetworking, IPV4 datagram, IPV6 packet format and advantages. Network Layer Protocols: ARP, RARP, IGMP and ICMP. Routing: Unicast Routing Protocols and Multicast Routing Protocols. Transport Layer: Process to Process Delivery, User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

Module-IV (06Hrs)

Domain Name System (DNS): Name Space, Domain Name Space, DNS in Internet, Resolution and Dynamic Domain Name System (DDNS), Remote logging, Electronic Mail (SMTP) and file transfer (FTP), Security services: Message confidentiality, integrity, authentication, non-repudiation, entity authentication, digital signature, key management

Text Books:

1. Data Communications and Networking, Behrouz A. Forouzan,(5th Edition) Tata McGraw-Hill.
2. Computer Networks, A. S. Tannenbum, D. Wetherall,(5th Edition) Prentice Hall, Imprint of Pearson.

Reference Book:

1. Network for Computer Scientists & Engineers, Zheng, Oxford University Press.
2. Computer Networks A system Approach, Larry L, Peterson and Bruce S. Davie, Elsevier.
3. Computer Networks, Natalia Olifer, Victor Olifer, Willey India.
4. Data and Computer Communications, William Stallings, Prentice Hall,Pearson.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

COMPILER DESIGN

Prerequisites	Theory of Computation, Programming Language, Data structures and Algorithms.
Course Outcomes	Understand basics of Compiler Design. Explain different translation Language. Understand DFA's, context free grammars, parse trees and abstract syntax trees. Understand various parsing techniques Understand the importance of Code Optimization, Code generation.

Module-1: 12 Hrs

Overview of the Translation Process: A Simple Compiler, Difference between interpreter, assembler and compiler. Overview and use of linker and loader, types of Compiler, Analysis of the Source Program, The Phases of a Compiler, Cousins of the Compiler, The Grouping of Phases, Lexical Analysis, Hard Coding and Automatic Generation Lexical Analyzers, Front-end and Back-end of compiler, pass structure. Lexical Analyzer: Introduction to Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzers, Finite Automata From a Regular Expression, Design of a Lexical Analyzer Generator, Optimization of DFA

Module-2: 12 Hrs

Parsing Theory: Top Down and Bottom up Parsing Algorithms, Top-Down Parsing, Bottom-Up Parsing, Operator-Precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators, Automatic Generation of Parsers. Syntax-Directed Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S-Attributed Definitions, L-Attributed Definitions, Syntax Directed Definitions and translation schemes. Error Recovery: Error Detection & Recovery, Ad-Hoc and Systematic Methods
Intermediate Code Generation: Different Intermediate Forms, Syntax Directed Translation Mechanisms and Attributed Mechanisms and Attributed Definition.

Module-3: 12 Hrs

Code Generation: Issues in the Design of a Code Generator, The Target Machine, Run-Time Storage Management, Basic Blocks and Flow Graphs, Next-Use Information, A Simple Code Generator, Register Allocation and Assignment, The DAG Representation of Basic Blocks, Peephole Optimization, Generating Code from DAGs, Dynamic Programming Code-Generation Algorithm, Code Generator Generators. Code Optimization: Global Data Flow Analysis, A Few Selected Optimizations like Command Sub Expression Removal, Loop Invariant Code Motion, Strength Reduction etc.

Text Books:

1. Compilers: Principles, Techniques and Tools By Aho, Lam, Sethi, and Ullman, Second Edition, Pearson, 2014
2. Compilers: Principles, Techniques and Tools By Aho, Sethi, and Ullman, Addison-Wesley, 1986

Reference Books:

1. Compiler Design in C By Allen I. Holub, Prentice-Hall/Pearson.
2. Advanced Compiler Design and Implementation By Muchnick, Morgan and Kaufmann, 1998

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

ARTIFICIAL INTELLIGENCE

Prerequisites	Basic Mathematics, Discrete Mathematics, Programming Languages, Algorithms
Course Outcomes	• Develop a basic understanding of AI building blocks presented in intelligent agents. • Choose an appropriate problem solving method and knowledge representation technique. • Design models for reasoning with uncertainty as well as the use of unreliable information. • Design and develop the AI applications in real world scenario.

Module 1 : (12 Hrs)

Introduction to AI, AI Technique, Level of the Model, Problem Spaces, and Search: Defining the Problem as a State Space Search, Production Systems, Problem Characteristics, Production System Characteristics, Issues in the Design of Search Programs. Heuristic Search Techniques: Generate-and-Test, Hill Climbing, Solution space search , Beam search Best-first Search, Problem Reduction, Constraint Satisfaction, Means-ends Analysis, Knowledge Representation: Representations and Mappings, Approaches to Knowledge Representation, Using Predicate Logic: First Order Logic (FOL),Representing Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution, Natural Deduction. Using Rules: Procedural Versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning, Matching, Control Knowledge.

Module 2: (12 Hrs)

Symbolic Reasoning Under Uncertainty: Introduction to Nonmonotonic Reasoning, Logics for Nonmonotonic Reasoning, Implementation Issues, Augmenting a Problem-solver, Depth-first Search, Breadth-first Search. Weak and Strong Slot-and-Filler Structures: Semantic Nets, Frames, Conceptual Dependency Scripts, CYC. Game Playing: The Minimax Search Procedure, Adding Alpha-beta Cutoffs, Iterative Deepening. Planning: The Blocks World, Components of a Planning System, Goal Stack Planning, Sussman's Anomaly, Nonlinear Planning Using Constraint Posting, Hierarchical Planning Other Planning Techniques. Understanding: What is Understanding, What Makes Understanding Hard?, Understanding as Constraint Satisfaction.

Module 3 : (12 Hrs)

Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Statistical Natural Language Processing, Spell Checking. Learning: Rote Learning, Learning by Taking Advice, Learning in Problem-solving, Learning from Examples: Induction, Explanation-based Learning, Discovery, Analogy, Formal Learning Theory, Neural Net Learning and Genetic Learning. Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

Text Book:

1. Elaine Rich, Kevin Knight, &Shivashankar B Nair, Artificial Intelligence, McGraw Hill,3rd ed.,2009

Reference Books:

1. Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PHI.,2010
2. S Kaushik, Artificial Intelligence, Cengage Learning, 1st ed.2011
3. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edition.

E COMMERCE AND ERP

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

Prerequisites	Computer Fundamentals
Course Outcomes	• Learn about electronic commerce development, deployment and utilization • Understand EDI operation and various E-business strategies.

Module –I (12 Hrs)

Introduction to E-commerce : Traditional Commerce and E-Commerce, Categories of E-Commerce, Framework of E-Commerce, Elements of E-Commerce: Network Infrastructure, Information Distribution Technology, Network Multimedia Content Publishing Technology, Security and Encryption, Payment Services, Business Service Infrastructure, Public Policy and Legal Infrastructure

Technology Infrastructure : Internet Protocols

Web Server Software : Web Server Basics, Software for Web Servers, Basic Functions of E-Commerce Software

Module–II (12 Hrs)

Security Threats to E-Commerce, Implementing E-Commerce Security

Selling on the Web: Revenue Models for Selling on the Web, Revenue Strategy Issues, Website Usability

Marketing on the Web: Web Marketing Strategies, Customer Relationship Intensity and Life-Cycle Segmentation, Advertising on the Web

Module–III (12 Hrs)

Business-to-Business Strategies : Electronic Data Interchange (EDI), Electronic Marketplaces

Web Auctions, Mobile Commerce, Virtual Communities

Payment Systems for E-commerce : Online payments Basics, Payment Cards, Electronic Cash, Electronic Wallets, Stored-Value Cards

Textbooks

1. Ecommerce, Gary P. Schneider, 4th Edition, Cengage Learning
2. Electronic Commerce: Framework Technologies & Applications, Bharat Bhasker, TMH

Reference Books

1. Electronic Commerce: A Manager's Guide, Kalakota & Whinston, Pearson
2. E-commerce: Concepts, models & strategies, C.V.S Murthy, Himalaya Publishing.

ADVANCED JAVA

Prerequisites	Programming languages C, C++, JAVA
Course Outcomes	● Understanding on Client-Server Architecture and TCP/IP programming ● Developing Web Applications using dynamic web pages and server-side programming through Servlets and JSPs ● Accessing Database from a Java Application using JDBC ● Developing distributed business applications using RMI

UNIT I (12 hrs)

Enterprise Application Architecture, Enterprise Java Technologies, Web Applications, Servlet Overview, Servlet API, Writing HelloWorld Program using Servlet, Servlet Life Cycle, Configuring Servlet in web.xml, Retrieving information from Request object, HTML form processing using Servlet, Servlet Initialization, Session tracking, Cookies, Database Access using Servlet, Error Handling, Servlet Collaboration, Forward verses Redirect.

Overview of JSP, JSP Advantages, JSP Application Models: JSPModel1 and Model 2 architectures, Life Cycle of a JSP page, JSP Elements, JSP Comments, Scripting in JSP, Directives, Implicit Objects, Action Tags, JSP and Java Beans, Introduction to JSTL, Introduction to JSP Expression Language.

UNIT II (10 hrs)

Introduction to JSF, Features, Benefits of JSF, JSF Architecture, JSF Elements, Request Processing Life Cycles, JSF HTML tags, JSF Core tags, Standard UI components, Managed Beans, Event handling, Page Navigation, convertors, validators, Expression Language, Using AJAX with JSF, sending AJAX Request;

UNIT III (14 hrs)

Enterprise JavaBeans Technology: EJB Component Architecture, Role of EJB& its life cycle, Types of Beans, Session Beans, Stateless and Stateful beans, Message Driven Bean, Life Cycle, Managing Transactions in EJB;

Understanding Java Persistence: Object Relational Mapping, Java Persistence API, Benefits, components of JPA, Entity, Entity manager, Persistence unit, Life cycle of Entity, Entity Relationships, querying entities, Java Persistence Query Language, performing CRUD operations using JPA; Introducing Hibernate.

Overview of SOA, Web Services, Types of Web Service, Building Web services with JAX-WS;

Text Books:

1. Java Server Programming (Java EE 7) Black Book, by DT Editorial Services, Dreamtech Press, 2015.

References:

1. Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito, "The Java EE 7 Tutorial", 5th Edition, Addison-Wesley Professional, Pearson India, 2014.
2. Advanced Java Technology by MT Savaliya, Dreamtech Press, 2015.
3. David Geary, Cay S. Horstmann, "Core Java Server Faces", Third Edition, 2010, Pearson Education, Inc. New Delhi.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

ADVANCED COMPUTER ARCHITECTURE

Prerequisites	Programming and Data structures, Basic knowledge of Computer organization.
Course Outcomes	• Understand the Concept of Parallel Processing and its applications • Implement the Hardware for Arithmetic Operations • Analyze the performance of different scalar Computers • Develop the Pipelining Concept for a given set of Instructions • Distinguish the performance of pipelining and non pipelining environment in a processor

Module 1: (12Hrs)

Processor Architecture : Evolution of Microprocessors, Instruction set processor design, Principles of processor performance, Instruction-level Parallelism, RISC and CISC architectures, Pipelining fundamentals, Arithmetic and instruction pipelining, Pipeline hazards, Minimizing pipeline stalls, Branch Prediction, superscalar and superpipelined architectures.

Module 2:(12Hrs)

Memory and I/O Architecture : Hierarchical memory technology; Multi-level caches, Data and Instruction caches, Cache optimizations, Memory Management hardware, I/O systems: Peripheral and Processor-Memory buses, Split transaction buses , USB.

Module 3:(12Hrs)

Multiprocessor Architecture :Basic multiprocessor architecture, Cache coherence, multithreaded processors, VLIW processor architectures. Array and vector processors. Case studies :MIPS architecture, Intel Series of processors, Pentium's Internally RISC and externally CISC, Hyper threading, SPARC and ARM processors.

Text Book

1. David A. Patterson and John L. Hennessy, Computer Organization and Design, Elsevier, Fourth Edition
2. John Paul Shen and MikkoLipasti, Modern Processor Design, Tata McGraw Hill.

Reference Books

1. DezsoSima, Terence Fountain, and Peter Kacsuk, Advanced Computer Architecture: A Design Space Approach, by Addison Wesley
2. John L. Hennessy & David A. Patterson, Computer Architecture, A Quantitative Approach 4th Edition, Morgan Kaufmann.
3. Hwang &Jotwani, Advance Computer Architecture, TMH

OPERATING SYSTEM LAB

Prerequisites	Working knowledge of computers, Programming languages
Course Outcomes	• Familiarize students with the architecture of Unix OS and Unix Commands. • Develop and debug, C programs created on UNIX platforms. • Implement CPU scheduling algorithms and Bankers algorithm used for deadlock avoidance and prevention. • Implement page replacement and memory management algorithms.

1. Basic UNIX Commands.
2. Linux Administrative commands.
3. UNIX Shell Programming.
4. Programs on process creation and synchronization, inter process communication including shared memory, pipes and messages.
5. Programs on process synchronization, (DinningPhilosopher problem / Cigarette Smoker problem / Sleeping barber problem)
6. Programs on UNIX System calls.
7. Simulation of CPU Scheduling Algorithms. (FCFS, RR, SJF, Priority, Multilevel Queuing)
8. Simulation of Banker's Algorithm for Deadlock Avoidance, Prevention
9. Program for FIFO, LRU, and OPTIMAL page replacement algorithm.
10. Programs on Multithreading

COMPUTER NETWORKS LAB

Prerequisites	Digital Electronics, Working knowledge of computers
Course Outcomes	• Analyze and simulate various networking protocols and mechanisms • Use tools to simulate real life network devices

1. Study of various network devices and network topologies
2. Implementation of cross-wired and straight-through cable using crimping tool
3. IP address configuration, network setup, troubleshooting and various network management commands
4. Building Class A, B and C networks using packet tracer software
5. Implementation of sub-netting concept using packet tracing software
6. Simulation of Hub Vs Switch networks by means of throughput and collision analysis using network simulator
7. Simulation of CSMA/CA Vs CSMA/CD using network simulator
8. Implementation of LSR and DVR routing protocols using network simulator
9. Installation of "ns2" in Linux environment
10. Basic wired and wireless topology in ns2 environment
11. Write a programme to retrieve the MAC address of a system using Address Resolution Protocol
12. Write a programme to find the class of a given IP address, subnet mask and address range of that subnet

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

COMPILER DESIGN LAB

Prerequisites	Theory of Computation, Programming Language, Data structures and Algorithms.
Course Outcomes	• Understand the working of lex and yac ccompiler for debugging of progs. • Understand and define the role of lexical analyser,use of regular expression and transition diagram. • Understand and use context free grammar and parse tree construction. • Learn and use new tools and technologies used for designing a compiler. • Develop progs for solving parse problems. • Learn how to write progs that execute faster.

(Any 10 Experiments)

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C/LEX language.
2. Write a program to identify whether a given line is a comment or not.
3. Write a program to recognize strings under 'a', 'a*b+', 'abb'.
4. Write a program to test whether a given identifier is valid or not.
5. Write a program to simulate lexical analyzer for validating operators.
6. Implement the lexical analyzer using JLex, flex or other lexical analyzer generating Tools.
7. Write a program for implementing the functionalities of predictive parser for the mini Language.
8. Write a program for constructing of LL (1) parsing
9. Write a program for constructing recursive descent parsing.
10. Write a program to implement LALR parsing.
11. Write a program to implement operator precedence parsing
12. Write a program to implement Program semantic rules to calculate the expression that takes an expression having digits, + and * and computes the value.
13. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree for the mini language.
14. Write a program to generate the code for the following three address code statements.
    A =B + C and W = X – Y
15. Write a program to generate code for the following three address code statements
    W = (A+B)*C

SIXTH SEMESTER

INTERNET AND WEB TECHNOLOGY

Prerequisites	Computer Networking
Course Outcomes	• Analyze a web page and identify its elements and attributes • Create web pages using HTML and Cascading Style Sheets • Build dynamic web pages using JavaScript (Client side programming). • Create XML documents and Schemas.

Module –I (12Hours)

Introduction, Evolution of Internet, WEB2.0, Understanding the WWW and the Internet, Emergence of Web, Protocols: TCP/IP, UDP, HTTP, Internet Addressing Scheme – Ipv4 & IPv6, Domain Name Server, Building Web Sites: Planning for designing Web pages, Model and structure for a Website, Developing Websites, Web Servers, Web Browsers.

HTML: Introduction, SGML, DTD (Document Type Definition). Basic HTML using images links, Lists, Tables and Forms, Frames for designing a good interactive website. HTML5: Migration, New Elements, Semantics, Canvas, SVG, Multimedia.

Module –II (12Hours)

CSS: Syntax, Class Selector, Id Selector, External, inline and Internal Style Sheets, div & span tags. DOM HTML DOM, inner HTML, Dynamic HTML (DHTML). CSS3: Rounded Corners, Border Images, Gradients, Shadows, 2D and 3D Transforms, Transitions, Animations, Box Sizing

Java Script: JAVA Script Programming Fundamentals, Statements, Expressions, Operators, Popup Boxes, Control Statements, Try…. Catch Statement, Throw Statement, Objects of JavaScript: Date object, array object, Boolean object, math object. Email and password validations. Writing Java Applets, Life cycle of applet, Design a login page using applets. Events and Event Handlers: General Information about Events, Defining Event Handlers, onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onMove, onReset, onSelect, onSubmit, onUnload

Module –III (12 Hrs)

What is XML – Basic Standards, Schema Standards, Linking & Presentation Standards, Standards that build on XML, Generating XML data, writing a simple XML File, Creating a Document type definition, Documents & Data, Defining Attributes & Entities in the DTD, Defining Parameter Entities & conditional Sections, Designing an XML data structure.

CGI/PERL: Introduction to CGI/Perl, Testing & Debugging Perl /CGI Script, Using Scalar variables and operators in Perl/CGI. PHP: Starting to script on server side, Arrays function and forms

INTERNET SECURITY & FIREWALLS: Types of Viruses, Client Server Security Threats, Data & Message Security, Encrypted Documents and Emails, Proxy Application Gateways, FIirewalls, AAA (Authentication, Authorization and Accounting).

Textbooks-

1. Web Warrior Guide to Web Design Technologies, Don Gosselin, Joel Sklar & others, Cengage Learning India, 2011
2. Programming the World Wide Web, Robert W Sebesta,8th edition, Pearson, 2015
3. Web Technologies, Uttam K Roy, Oxford,2010

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

SOFTWARE ENGINEERING

Prerequisites	Knowledge of computer, problem Solving Skills, Object Oriented Programming Concepts
Course Outcomes	• Develop a thorough understanding of software development lifecycle principles • Able to design and plan software solutions to problems using an object-oriented strategy • Able to develop and apply testing strategies for software applications • Develop an estimation of the cost, quality, and management issues involved in software construction

Module I (12 hours)

Software Process Models: Software Product, Software crisis, Handling complexity through Abstraction and Decomposition, Overview of software development activities, Process Models, Classical waterfall model, iterative waterfall model, prototyping mode, evolutionary model, spiral model, RAD model, V model, Agile models: Extreme Programming, and Scrum.

Software Requirements Engineering: Requirement Gathering and Analysis, Functional and Non-functional requirements, Software Requirement Specification (SRS), IEEE 830 guidelines, Decision tables and Decision trees.

Module II (12 hours)

Structured Analysis & Design: Overview of design process: High-level and detailed design, Cohesion and coupling, Modularity and layering. **Function–Oriented software design:** Structured Analysis using DFD Structured Design using Structure Chart, Basic concepts of Object Oriented Analysis & Design. User interface design, Command language, menu and iconic interfaces **Coding and Software Testing Techniques:** Coding, Code Review, documentation. **Testing:** - Unit testing, Black-box Testing, White-box testing, Cyclomatic complexity measure, coverage analysis, mutation testing, Debugging techniques, Integration testing, System testing, Regression testing.

Module III (12 hours)

Software Reliability and Software Maintenance: Basic concepts in software reliability, reliability measures, reliability growth modeling, Quality SEI CMM, Characteristics of software maintenance, software reverse engineering, software reengineering, software reuse. **Software Size Metrics:** LOC Function Point, **COCOMO Model:** Basic COCOMO Model, Intermediate COCOMO model, Complete COCOMO model. Emerging Topics: Client-Server Software Engineering, Service-oriented Architecture (SOA), Software as a Service (SaaS), CORBA.

Text Book:

1. Fundamentals of Software Engineering, Rajib Mall, PHI, 2014.

Reference Books:

1. Software Engineering, A Practitioner's Approach, Roger S. Pressman, TMG Hill.
2. Software Engineering, I. Sommerville, 9th Ed. , Pearson Education.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

DATA MINING

Prerequisites	Data Structures and Algorithms, Database, SQL
Course Outcomes	• Understand the fundamentals of data warehousing and data mining • Design data warehouses and define specific OLAP operations for analysis • Apply data mining techniques like classification, prediction, clustering • Gain knowledge about complex data types and spatial data mining.

Module - I (12 Hours)

Introduction to Data warehousing: Definition and Characteristic, Need for data warehousing, Evolution of Decision support System, Building blocks of data warehouse, data warehouses and data marts, metadata in the data warehouse, Data warehousing Architecture, Data warehousing implementation, Business and data warehouse

Data Warehouse Modelling and Design: Multidimensional data model, Data cube, Schemas for multidimensional data models (Star, Snowflake, Fact Constellation), OLAP, OLAP Operations, OLAP Models(ROLAP, MOLAP, HOLAP), OLAP vs OLTP, Benefits of Data Warehousing

Module - II (12 Hours)

Introduction to Data Mining: KDD Process, Data mining Functionalities, Classification of data mining systems, data mining task primitives, Integration of data mining system with data warehouse, Data Preprocessing (data summarization, data cleaning, data integration and transformation, data reduction, data discretization)

Association Rule Mining: Mining frequent patterns, associations, correlations (market basket analysis), Frequent Itemset Mining, (Apriori algorithm, FP-Growth), Correlation Analysis (Chi-square, Lift), Kinds of association rules

Module - III (12 Hours)

Classification: Classification vs Prediction, issues, Decision tree induction, Attribute Selection Measures, Tree Pruning, Rule based classification, classification by Back Propagation, Bayseian Classification, Support Vector Machines

Cluster Analysis: Data in cluster analysis, Categorization of clustering methods, partitioning methods (k-means, k-medoids), hierarchical methods(AGNES, DIANA, BIRCH),density based methods (DBSCAN, OPTICS), Outlier Analysis

Text Books:

1. Data Mining: Concepts and techniques: Han, Camber and Pei, Elsevier (3rd Edition).
2. Data Mining & Data Warehousing Using OLAP: Alex & Stephen, McGraw Hill

Reference books:

1. Data Mining Techniques and Applications by Hongbo Du, Cengage
2. Data Mining: Arun Pujari, University Press
3. Data Mining –a Tutorial based primer by R.J.Roiger, M.W.Geatz, Pearson Education.
4. Data Warehousing: ReemaThareja, Oxford University Press
5. Data warehousing Fundamentals: PaulrajPonniah, Willey India.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

DATA ANALYTICS

Prerequisites	Linear algebra, calculus, probability theory, statistics, and programming languages
Course Outcomes	• Demonstrate the ability to think critically in making decisions based on data and deep analytics. • Demonstrate the ability to use technical skills in predicative and prescriptive modeling to support business decision-making. • Demonstrate effective communication skills that facilitate the effective presentation of analysis results. • Execute real-time analytical methods on streaming datasets to react quickly to customer needs

MODULE I: 12 Hrs
Introduction: Data Analytics, Data Mining and Knowledge Discovery, Data and Relations: The Iris Data Set, Data Scales, Set and Matrix Representations, Relations, Dissimilarity Measures, Similarity Measures, Sequence Relations, Sampling and Quantization, Data Preprocessing: Error Types, Error Handling, Data Transformation.

MODULE II: 12 Hrs
Data Visualization: Principal Component Analysis, Multidimensional Scaling, Sammon Mapping, Correlation: Linear Correlation, Correlation and Causality, Chi-Square Test for Independence, Regression: Linear Regression, Linear Regression with Nonlinear Substitution, Robust Regression, Neural Networks, Radial Basis Function Networks, Cross-Validation.

MODULE III: 12 Hrs
Forecasting: Recurrent Models, Autoregressive Models, Classification: Classification Criteria, Naive Bayes Classifier, Linear Discriminant Analysis, Support Vector Machine, Nearest Neighbor Classifier, Learning Vector Quantization, Decision Trees, Clustering: Cluster Partitions, Sequential Clustering, Prototype-Based Clustering, Fuzzy Clustering, Relational Clustering, Cluster Tendency Assessment, Cluster Validity, Some Optimization Methods: Optimization with Derivatives, Gradient Descent, Lagrange Optimization.

TEXT BOOKS:

Thomas A. Runkler, Data Analytics Models and Algorithms for Intelligent Data Analysis, Springer, 2012.

COMPUTER VISION

Prerequisites	Computer Graphics
Course Outcomes	• Identify basic concepts, terminology, theories, models and methods in the field of computer vision • Describe known principles of human visual system • Describe basic methods of computer vision related to multi-scale representation, edge detection

Module-I (12 hours)
Digital Image Fundamentals: Basic Concepts, Imaging geometry, translation, scaling, rotation, Image formation, Geometric Camera Models, Image digitization, Image types, Color Images, color models, Digital Image Properties: metrics and topological properties of digital images, histograms, visual properties.
Image pre-processing: Pixel brightness transformation, Local pre-processing.

Module-II (12 hours)
Image segmentation: Fundamentals, point, line detection, basic edge detection techniques, Hough transform, Thresholding, basic global threholding, optimal thresholding using Otsu's method, multi-spectral thresholding, Region based segmentation, region growing, region splitting and merging.
Shape representation and descriptors: Region identification, Contour or boundary-based representation and descriptors, chain codes, boundary length, curvature, bending energy, signature, Fourier descriptor.

Module-III (12 hours)
Region-based representation and descriptors, area, Euler's number, eccentricity, elongatedness, rectangularity, direction, compactness, moments, covex hull Object and Pattern Recognition: Pattern and pattern classes, Matching, minimum distance or nearest neighbor classifier, matching by correlation, Optimum statistical classifier, Neural network classifier.

TEXT BOOK
1. Milan Sonka, Vaclav Hlavac and Roger Boyele, Image processing, analysis, and machine vision. 3e, Cengage Learning, 2014.

REFERENCE BOOKS:
1. Rafael C. Gonzalez and Richard E. Woods. "Digital image processing"Pearson Education
2. Computer Vision A modern approach, David A. Forsyth and Jean Ponce, Pearson Education

CLOUD COMPUTING

Prerequisites	Computer Networking, Operating System
Course Outcomes	• Define Cloud Computing and memorize the different Cloud service and deployment models • Describe importance of virtualization along with their technologies. • Use and Examine different cloud computing services • Analyze the components of open stack & Google Cloud platform and understand Mobile Cloud Computing • Describe the key components of Amazon web Service • Design & develop backup strategies for cloud data based on features.

Module-1 : (12 hrs)

Introduction: Cloud-definition, benefits, usage scenarios, Functioning of Cloud Computing – Cloud Architecture – Types of Clouds – Business models around Clouds – Major Players in Cloud Computing – issues in Clouds , Risks Involved in Cloud Computing.

Cloud Services: Types of Cloud services: Software as a service – Platform as a Service – Infrastructure as a Service – database as a Service – Monitoring as a Service – Communication as services, Service providers – Google, Amazon, Microsoft Azure, IBM, Salesforce

Cloud Service Administration- Service Level Agreements and Monitoring-Support Services- Accounting Services, Resource Management- IT Security- Performance Management- Provisioning- Service Management, Untangling Software Dependencies.

Module-2: (12 hrs)

Collaborating Using Cloud Services: Email Communication over the Cloud – CRM Management – Project Management – Event Management – Task Management – Calendar – Schedules – Word Processing – Presentation – Spreadsheet – Databases – Desktop – Social Networks and Groupware, Work Loan Management in Cloud.

Virtualization For Cloud: Need for Virtualization – Pros and cons of Virtualization – Types of Virtualization – System Vm, Process VM, Virtual Machine monitor – Virtual machine properties – Interpretation and binary translation, HLL VM – Hypervisors – Xen, KVM, VMWare, Virtual Box, Hyper-V.

Module-3: (12 hrs)

Data & Cloud Storage: Enterprise Data Storage(SAN,NAS),Cloud File System,Cloud Data stores & Data management for cloud storage.

Security, Standards and Applications: Security in Cloud: Cloud security challenges – Software as a Service Security, Common Standards: The Open Cloud Consortium – The Distributed Management Task Force – Standards for application Developer – Standards for Messaging – Standards for Security, End user access to cloud computing, Mobile Internet devices and the cloud.

Cloud Computing Platforms & tools: Eucalyptus – Nimbus – Open Nebula, CloudSim,Apache,Hadoop,Map Reduce

Text Books:

1. John Rittinghouse and James Ransome, "Cloud Computing, Implementation, Management and Strategy", CRC Press, 2009.
2. Cloud Computing Principles & Paradigms By Buyya, Brobery & Goscinni(Wiley).
3. Cloud Computing By Srinivasan & Suresh(Pearson).

References:

1. Michael Miller, "Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate", Que Publishing, August 2008.
2. James E Smith and Ravi Nair, "Virtual Machines", Morgan Kaufmann, 2006.
3. David E. Y. Sarna, "Implementing and Developing Cloud Application", CRC press 2011.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

COMPUTER GRAPHICS

Prerequisites	Basic Mathematics, Programming Languages, Algorithms
Course Outcomes	• Understand the basics of computer graphics, different graphics systems and applications of computer graphics. • Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis. • Use of geometric transformations on graphics objects and their application in composite form. • Extract scene with different clipping methods and its transformation to graphics display device.

Module – I (10 hours)

Overview of Graphics System: Video Display Units, Raster-Scan and Random Scan Systems, Graphics Input and Output Devices. Output Primitives: Line drawing Algorithms: DDA and Bresenham's Line Algorithm, Circle drawing Algorithms: Midpoint Circle Algorithm and Bresenham's Circle drawing Algorithm. Two Dimensional Geometric Transformation: Basic Transformation (Translation, Rotation, Scaling) Matrix Representation, Composite Transformations, Reflection, Shear, Transformation between coordinate systems.

Module – II (10 hours)

Two Dimensional Viewing: Window-to- View Port Coordinate Transformation. Line Clipping (Cohen-Sutherland Algorithm) and Polygon Clipping (Sutherland-Hodgeman Algorithm) Aliasing and Antialiasing, Half Toning, Thresholding, Dithering. Polygon Filling: Seed Fill Algorithm, Scan line Algorithm. Two Dimensional Object Representations: Spline Representation, Bezier Curves, B-Spline Curves. Fractal Geometry: Fractal Classification and Fractal Dimension.

Module – III (8 hours)

Three Dimensional Geometric and Modeling Transformations: Translation, Rotation, Scaling, Reflections, shear, Composite Transformation. Projections: Parallel Projection, Perspective Projection. Visible Surface Detection Methods: Back-Face Detection, Depth Buffer, A- Buffer, Scan- Line Algorithm, Painters Algorithm.

Module – IV (8 hours)

Illumination Models: Basic Models, Displaying Light Intensities. Surface Rendering Methods: Polygon Rendering Methods: Gouraud Shading, Phong Shading. Computer Animation: Types of Animation, Key frame Vs. Procedural Animation, Methods of Controlling Animation, Morphing. Introduction to Virtual Reality and Augmented Reality.

Textbook:

1. Computer Graphics, D. Hearn and M.P. Baker (C Version), Pearson Education, Seventh Impression 2008

Reference Books:

1. Computer Graphics Principle and Practice, J.D. Foley, A. Dam, S.K. Feiner, Addison Wesley, 2nd Edition.
2. Procedural Elements of Computer Graphics, David Rogers, TMH.
3. Computer Graphics: Algorithms and Implementations, D.P Mukherjee, D. Jana, PHI.
4. Computer Graphics, Z. Xiang, R. A. Plastock, Schaum's Outlines , McGrow Hill.

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

SOFT COMPUTING

Prerequisites	Evolutionary Computation, Probabilistic Reasoning
Course Outcomes	• Apply soft computing methodology for a particular problem • Apply neural networks to pattern classification and regression problems • Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems • Apply genetic algorithms to combinatorial optimization problems

Module I: (12 hrs)

Fundamentals of Neural Network:- Introduction to Soft Computing, Neural Network, Neural Network Application, Neural Network Architecture

Simple Neural Network:- Perceptron, McCulloh Pitt Model, Adaline, Madaline, Hebb's Net, Back propagation Neural Network, Auto associative and Hetero associative network, Kohen self-organizing map, Artificial Resonance Theory

Module II: (8 hrs)

Fuzzy Logic: - : Fuzzy Logic: Crisp and Fuzzy sets, membership functions, Basic operations on fuzzy sets, Properties of fuzzy sets, Fuzzy relations, Propositional and predicate logic, fuzzy mapping rules and implications, Fuzzy models, Applications.

Module III: (10 hrs)

Nature Inspired Algorithms: Introduction, Genetic algorithms, Differential evolution, Particle swarm optimization, Ant colony optimization, Bacteria Foraging Optimization, Cuckoo search.

Module IV: (6 hrs)

Hybrid Systems: Integration of neural networks, fuzzy logic and genetic algorithms.

Text/ Reference books:

1. S. Rajasekaran and G.A. Vijaylakshmi Pai, Neural Networks Fuzzy Logic, and Genetic Algorithms, Prentice Hall of India, 2003.
2. J.S.R. Jang, C.T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall of India, 2004.

Reference books:

1. N.P. Padhy and S.P. Simon, Soft Computing: With Matlab Programming, Oxford University Press, 2015.
2. Xin-She Yang: Nature Inspired Optimization Algorithms ,Elsevier,2014
3. D. E. Goldberg, Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley, 1989.
4. Lauren Fausett: Fundamentals of Neural network-Architecture, Algorithm, Application , Pearson,2004

INTERNET AND WEB TECHNOLOGIES LAB

Prerequisites	Programming Languages
Course Outcomes	• Design and implement dynamic websites with good aesthetic sense of designing • Create a static website using HTML and add dynamic functionality to it by using java Script. • Create dynamic website on real world problems.

Experiment 1 : Basic HTML using images links, Lists.
Experiment 2 : Basic HTML using Tables and frames.
Experiment 3 : Design the web page using form elements.
Experiment 4 : Design a web page by using HTML 5
Experiment 5 : Use different types of styles in a page. Use inline, internal and external style sheets.
Experiment 6 : Design a web page by using CSS3
Experiment 7 : Embedding the JavaScript in to Web pages. Usage of concepts and functions, array and objects.
Experiment 8 : Form validation using JavaScript
Experiment 9 : DTD and XML implementations in web designing concepts
Experiment 10: Create the small web site using web2.0 concepts.

SOFTWARE ENGINEERING LAB

Prerequisites	Knowledge of computer, problem Solving Skills, Object Oriented Programming Concepts
Course Outcomes	• Classifying the requirements and preparing software requirement documents for analyzing the projects. • Understanding the different design techniques • Implementation of UML Modeling, Use case design • Learning the use of tools like Rational Rose or Argo UML (opensource)

Experiment 1: Develop requirements specification for a given problem. The requirements Specification should include both functional and non-functional requirements
Experiment 2: Develop DFD Model (Level 0, Level 1 DFD) of the sample problem
Experiment 3: Develop DFD model (Level 2) and data dictionary of the sample problem
Experiment 4: Develop UML Use case model for a problem (Use of a CASE tool any of Rational rose, Argo UML, or Visual Paradigm etc. is required)
Experiment 5: Develop Class diagrams
Experiment 6: Develop Sequence Diagrams
Experiment 7: Develop Collaboration Diagrams
Experiment 8: Develop Activity Diagram Diagrams
Experiment 9: Develop State Chart Diagrams
Experiment 10: Develop code for the developed class model using Java & testing

SEVENTH SEMESTER
MOBILE COMPUTING

Prerequisites	Computer Network, Data Communication, Operating System
Course Outcomes	• Explain the basic concepts of wireless network and wireless generations. • Demonstrate the different wireless technologies such as CDMA, GSM, GPRSetc • Describe and judge the emerging wireless technologies standards such as WLL, WLAN, WPAN, WMAN. • Explain the design considerations for deploying the wireless network infrastructure.

Module - I (12 hours)

Introduction: Three Tier Architecture Mobile Computing Architecture, Evolution of Wireless Technology

Cellular System: Cell, Cluster, Cell Splitting, Frequency Reuse, Channel Assignment Strategies, Components of Cellular System, Operation of Cellular System

Personal Communications Services (PCS): PCS Architecture, mobility management.

Global System for Mobile Communication (GSM): Overview, Architecture, Network signaling, Channels, Mobility Management.

General Packet Radio Services (GPRS): Architecture, GPRS Interfaces, Network Protocols

Module - II (12 hours)

Wireless LAN (WLAN): Application, Requirement, IEEE 802.11(Ad-hoc Mode, Infrastructure Mode, Protocol Architecture), Bluetooth (Piconet, Scatternet, Protocol Stack, Bluetooth Profile)

Wireless Application Protocol (WAP): WAP Gateway and Protocols, wireless mark up Languages (WML),

Wireless Local Loop (WLL): WLL Configuration, Architecture, WLL Technologies.

IMT 2000: Vision, IMT-2000 Family, W-CDMA and CDMA 2000

Module - III (12 hours)

Satellite System: Introduction, Types of Satellite System (HEO, MEO, LEO), Satellite System Architecture, Case Studies: IRIDIUM, GLOBALSTAR and ICO

Virtual Private Network (VPN): Features, Goals, Working, Remote Access VPN, Site to Site VPN, VPN Protocols, Requirements

Mobile IP & Cellular IP: Goals & Working, **Mobile OS:** Windows CE, Windows Mobile OS, Symbian OS

Text Books:

1. Mobile Communication: J. Schiller, 2ND Edition, Pearson Education
2. Mobile Computing: AsokeTalukdar, 2nd Edition, TMH.

Reference Books:

1. Fundamentals of Mobile Computing, Prashanta Kumar Patnaik and Rajib Mall, PHI, 2nd Edition, 2015
2. Mobile Computing: P.K. Patra, S.K. Dash, 2nd Edition, Scitech Publications.
3. Mobile Computing, Raj Kamal, 2nd Edition, Oxford University Press

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

REAL TIME SYSTEMS

Prerequisites	Operating Systems, Computer Networks, Database
Course Outcomes	• Develop real-time algorithm for task scheduling. • To understand the working of real-time operating systems and real-time database. • To work on design and development of protocols related to real-time communication.

MODULE-1 (12Hrs)
Introduction: What is real time, Applications of Real-Time systems, A basic model of Real-time system, Characteristics of Real-time system, Safety and Reliability, Types of Real-time tasks, timing constraints, Modeling timing constraints Real-Time Task Scheduling: Some important concepts, Types of Real-time tasks and their characteristics, Task scheduling, Clock-Driven scheduling, Hybrid schedulers, Event Driven scheduling, Earliest Deadline First (EDF) scheduling, Rate monotonic algorithm (RMA). Issues Associated with RMA.

MODULE-2 (12Hrs)
Handling Resource Sharing and dependencies among Real-time Tasks: Resource sharing among real-time tasks. Priority inversion. Priority Inheritance Protocol (PIP), Highest Locker Protocol (HLP). Priority Ceiling Protocol (PCP). Different types of priority inversions under PCP. Important features of PCP. Some issues in using a resource sharing protocol. Handling task dependencies. Scheduling Real-time tasks in multiprocessor and distributed systems: Multiprocessor task allocation, Dynamic allocation of tasks. Fault tolerant scheduling of tasks. Clock in distributed Real-time systems, Centralized clock synchronization

MODULE-3 (12Hrs)
Commercial Real-time operating systems: Time services, Features of a Real-time operating system, Unix as a Real-time operating system, Unix-based Real-time operating systems, Windows as a Real-time operating system, POSIX-RT, A survey of contemporary Real-time operating systems. Benchmarking real-time systems. Real-time Databases: Example applications of Real-time databases. Review of basic database concepts, Real-time databases, Characteristics of temporal data. Concurrency control in real-time databases. Commercial real-time databases. Real time Communication: Basic concepts, Examples of applications, Real-time communication in a LAN and Real-time communication over packet switched networks.

Text Book:
Real-time Systems Theory and Practice by Rajib Mall, Pearson Publication, 2008.

Reference BOOK:
1. Jane W. S. Liu, Real-Time Systems, Pearson Education, 2000.
2. C.M. Krishna and K.G. Shin, Real-Time Systems, TMH.

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

WIRELESS SENSOR NETWORKS

Prerequisites	Computer Networks, Operating system, Mobile Computing
Course Outcomes	• Understand the basic WSN technology and supporting protocols, with emphasis placed on standardization basic sensor systems and provide a survey of sensor technology • Learn key routing protocols for sensor networks and main design issues • Learn transport layer protocols for sensor networks, and design requirements • Understand the Sensor management , sensor network middleware, operating systems

Module I : (12hrs)
Introduction: Introduction to Wireless Sensor Networks, Node architecture, Advantages of Sensor Networks, Application of Sensor Networks, Challenges and Constraints.
Node Architecture: Sensing Subsytem, Processor Subsystem, Communication Interfaces
Operating System: Functional Aspects, Non-functional Aspects, Prototypes(Tiny OS, SOS, Contiki, Lite OS)

Module II : (12 hrs)
Physical Layer: Basic components, Source and Channel Encoding, Modulation, Signal Propagation.
MAC Layer: Wireless MAC Protocols (CSMA, MACA, MACAW), Characteristics of MAC protocols in Sensor Networks, Contention-Free MAC protocols(TRAMA, YMAC, LEACH), Contention-Based MAC protocols(PAMAS, SMAC, TMAC), Hybrid MAC protocols.
Network Layer: Classification of Routing Protocol, Routing metrics, Flooding and gossiping, Data-Centric routing (SPIN, Directed Diffusion, Gradient), Proactive routing (DSDV, OLSR), On-Demand routing (AODV, DSR), Hierarchical routing, Location-Based routing (UNICAST, MULTICAST, GAF), QoS-Based routing protocols

Module III: (12hrs)
Localization: Ranging Techniques, range based localization, Range Free Localization, event Based Localization
Time Synchronization: Reasons and challenges for time synchronization, Basics of time synchronization, Time synchronization protocols – Receiver Broadcast Synchronization, Timing-Sync Protocol for Sensor Networks and Flooding Time Synchronization Protocol.

Text Books :
1. Fundamentals of Wireless Sensor Network: Theory and Practice: Waltenegus Dargie and Christian Poellabauer, Wiley Publication, 2010.
2. Networking Wireless Sensors: BhaskarKrismachari, Cambridge University Press
References Books:
1. Wireless Sensor Networks: An Information Processing Approach- by Feng Zhao, Leonidas Guibas , Morgan Kaufmann Series in Networking 2004
2. Wireless Sensor Networks: Edited by C.S Raghavendra, Krishna M, Sivalingam, TaiebZnati, Springer

Abbreviations Used: L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

SOFTWARE PROJECT MANAGEMENT

Prerequisites	Software Engineering
Course Outcomes	• Estimate project cost and perform cost-benefit evaluation among projects • Perform project scheduling, activity network analysis and risk management • Apply schedule and cost control techniques for project monitoring including contract management. • Apply quality models in software projects for maintaining software quality and reliability.

Module I: Project Evaluation and Planning (12Hrs)
Introduction to Software Project Management : Software Projects vs Other Types of Projecr, Activities in Software Project Management, Evaluation of Individual Projects(Cost–Benefit Analysis, Cash Flow Forecasting), Cost Benefit Evaluation Techniques, Risk Evaluation , Software Effort Estimation : COCOMO 2, Cost Estimation, Staffing pattern, Effect of schedule compression, Capers Jones estimating rules of thumb, Activity Planning : Objectives of Activity Planning, Sequencing and Scheduling Activities, Network Planning Models, Critical path analysis,.

Module 2: Monitoring and Control (12Hrs)
Collecting Data, , project termination review, Visualizing Progress, Cost Monitoring, Earned Value analysis, Change Control, Software Configuration Management (SCM), Managing Contracts : Types Of Contracts, Stages In Contract Placement, Typical Terms of A Contract, Contract Management and Acceptance.

Module 3: Quality Management and People Management (12Hrs)
Introduction, Understanding Behavior, Organizational Behaviour, Selecting The Right Person For The Job, Motivation, The Oldman – Hackman Job Characteristics Model, Stress, Health And Safety, Working in Teams : Decision Making, Organization and team structures, Leadership, ISO and CMMI models, Testing and Software reliability.

Text Book :
1. Bob Hughes, Mike Cotterell, "Software Project Management", Fifth Edition, Tata McGraw Hill, 2011.

Reference Books :
1. Royce, "Software Project Management", Pearson Education, 1999.
2. Robert K. Wysocki, Effective Software Project Management, Wiley, 2009.

INFORMATION RETRIEVAL

Prerequisites	Data Structures and Files Database management systems, Query Processing, Data Mining
Course Outcomes	• Understand the concept of Information retrieval. • Deal with storage and retrieval process of text and multimedia data. • Evaluate performance of any information retrieval system. • Understand importance of recommender system. • Understand concept of multimedia and distributed information retrieval.

Module-I: 12Hours

Basic Concepts of IR, Data Retrieval & Information Retrieval, IR system block diagram.IR Models: Boolean Model, Vector Model

Storage: Inverted file, Suffix trees &suffix arrays, Signature Files, Scatter storage or hash addressing, Clustered files.

Searching strategies: Boolean Search, Serial search, cluster based retrieval, Query languages, Types of queries, Patterns matching, structural queries.

Module-II: 12Hours

Index Construction, Dynamic indexing index compression, vector space retrieval, similarity search Probabilistic IR: Probabilistic information retrieval, Language model of information retrieval, Bottom-up and top-down parsing paradigms, clustering and visualization via embedding

Module-III: 12Hours

Learning in IR: Supervised Learning, Evaluating text classifiers, Nearest neighbors learners, Bayesian Learners, Hypertext Classification, Semi supervised learning

Recommender Systems: Collaborative Filtering and Content Based Recommendation of Documents and Products, Information Extraction and Integration: Extracting Data from Text. Semantic Web, Collecting and Integrating Specialized Information on the web.

Text Books

1. Yates & Neto, "Modern Information Retrieval", Pearson Education,ISBN81-297-0274-6.
2. Heiner Stuckenschmidt, Frank van Harmelen, “Information Sharing on the Semantic Web”, Springer International Edition, ISBN3-540-20594-2.
3. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze “Introduction to Information Retrieval ”, Cambridge University Press, ISBN 978-0-521-86571-5

Reference Books

1. Markleven, “Introduction to search engines and web navigation”, John Wiley and sons
2. V. S. Subrahamanian, Satish K. Tripathi “Multimedia information System”, Kulwer Academic Publisher.
3. Chabane Djeraba, ”Multimedia mining A high way to intelligent multimedia documents”, Kulwer Academic
4. Stefan Buttcher, Charles L.A. Clarke, Gordon V. Cormack, Information Retrieval Implementing and Evaluating Search Engines, The MIT Press, Cambridge, Massachusetts London, England,2010.

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

FAULT TOLERANT SYSTEMS

Prerequisites	Digital electronics, Computer Architecture, operating systems, Computer networks
Course outcome	Learn and design fault tolerant systems which are need of the future

Module-I : 12 Hrs
BASIC TERMS:
Definition of fault tolerance, Redundancy, Applications of fault-tolerance, Fundamentals of dependability.
Attributes: Reliability, availability, safety, Impairments: faults, errors and failures, Means: fault prevention, removal and forecasting
DEPENDABILITY EVALUATION:
Common measures: failures rate, mean time to failure, mean time to repair, etc. Reliability block diagrams, Markov processes

Module-II: 12 Hrs
REDUNDANCY:
Hardware redundancy, Redundancy schemes, Evaluation and comparison, Applications, Information redundancy, Codes: linear, Hamming, cyclic, unordered, arithmetic, etc., Encoding and decoding techniques, Applications, Time redundancy

Module-III : 12 Hrs
PROGRAMMING:
Software fault tolerance, Specific features, Software fault tolerance techniques: N-version programming, recovery blocks, self-checking software, etc.

Text Books
1. Anderson, T., and P.A. Lee, Fault-Tolerant Principles and Practices, Prentice-Hall
2. Hwang, K., and F.A. Briggs, Computer Architecture and Parallel Processing, McGraw-Hill.
3. Jalote, P.Fault-Tolerance in Distributed Systems, ISBN 0-13-301367-7, Prentice-Hall,

Reference Books
1. Johnson, B.W., Design and Analysis of Fault-Tolerant Systems, Addison Wesely
2. Leveson, Nancy G., Safeware, system safety and computers, Addison Wesely.
3. Pradhan, D.K., Fault-Tolerant Computing — Theory and Techniques, (2 Volumes), Prentice-Hall.
4. Pradhan, Dhiraj K., Fault-Tolerant Computer System Design, ISBN 0-13-057887-8, Prentice-Hall

Entrepreneurship Development (3-0-0)

Prerequisites:

1. Organizational Behaviour.
2. English.

Module 1: (06 Hours)

Entrepreneurship: Concept of Entrepreneurship and Intrapreneurship, Types of Entrepreneur, Nature and Importance, Entrepreneurial Motivation and Achievement, Entrepreneurial Personality & Traits and Entrepreneurial Skills.

Module 2: (08 Hours)

Entrepreneurial Environment, Identification of Opportunities, Converting Business, Opportunities into reality. Start-ups and business incubation, Skill Development. Setting up a Small Enterprise. Issues relating to location, Environmental Problems and Industrial Policies and Regulations.

Module 3: (08 Hours)

Basics of Accounting, Terms: Assets, Liabilities, Equity, Revenue, Expense, Working capital, Marketing Mix and STP.

HRM: Concepts and Function, Labour Laws- Factories Act, Organizational sup- port services - Central and State Government, Incentives and Subsidies.

Module 4: (08 Hours)

Sickness of Small-Scale Industries, Causes and symptoms of sickness, cures of sickness, Role of Banks and Government in reviving sick industries.

Text Books:

1. Entrepreneurship Development and Management, Vasant Desai, HPH.
2. Entrepreneurship Management, Bholanath Dutta, Excel Books.
3. Entrepreneurial Development, Sangeeta Sharma, PHI.
4. Entrepreneurship, Rajeev Roy, Oxford University Press.

EIGHTH SEMESTER
MACHINE LEARNING

Prerequisites	Probability, Linear Algebra, Programming Languages
Course outcome	• Identify the characteristics of datasets and compare the trivial data and big data for various applications. • Implement machine learning techniques and computing environment that are suitable for the applications under consideration. •Solve problems associated with batch learning and online learning, • Integrate machine learning libraries and mathematical and statistical tools with modern technologies like hadoop and mapreduce.

Module 1:(9 hrs)
Introduction: well-posed learning problems, designing a learning system, perspectives and issues in machine learning, **concept learning and the general-to-specific ordering**: Introduction, A concept learning task, concept learning as search, version spaces and the CANDIDATE-ELIMINATION, Inductive Bias **Decision tree learning:** Introduction, decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning.
Module 2:(11 hrs)
Artificial Neural Networks: Introduction, biological motivation, neural network representation, appropriate problem for neural network learning, Perception, multilayer networks and the back propagation algorithm,
Linear Regression, Support vector machine, kernel function and kernelSVM.
Module 3:(8 hrs)
Bayesian learning: Introduction, Bayes theorem, Bayes theorem and concept learning, maximum likelihood and least-squared error hypotheses, minimum description length principle, Bayes optimal classifier, Gibbs algorithm, Naive Bayes classifier, example to illustrate Naïve Bayes classifier
Instance-Based Learning: Introduction, K-Nearest Neighbor Learning, Radial Basis Functions **Module 4:(8 hrs)**
Clustering: k-means, adaptive hierarchical clustering, Gaussian mixture model, computational learning theory, PAC learning model, sample complexity, VC dimension, ensemble learning.
Learning Set of Rules: Introduction, sequential covering algorithms, learning rule sets, learning First order Rules, learning sets of first order rules, inverting resolution

TEXT BOOK
1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.

REFERENCE BOOKS
1. Introduction to Machine Learning Edition 2, by EthemAlpaydin
2. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press,2012.

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

EMBEDDED SYSTEM

Prerequisites	Digital electronics, Computer Architecture, operating systems, Computer networks
Course outcome	• Describe the differences between the general computing system and the embedded system, also recognize the classification of embedded systems.. • Aware of the architecture of the ATOM processor and its programming aspects (assembly Level) • Aware of interrupts, hyper threading and software optimization. • Design real time embedded systems using the concepts of RTOS.

Module I (12hrs)

Introduction: Embedded System, Application and characteristics of embedded systems, Overview of Processors and hardware units in embedded system, embedded software in a system, Examples of Embedded system. Design metrices of Embedded system.

ARM: ARM Microcontroller, ARM pipeline, Instruction Set Architecture ISA: Registers, Data Processing Instructions, Data Transfer Instructions, Multiplications instructions, Software interrupt, Conditional execution, branch instruction, Swap instruction, THUMB instructions.

Module II (12hrs)

Device and Device drivers: I/O devices, Timer and Counting devices, Serial peripheral interfaces,, IIC, RS232C,RS422,RS485, Universal serial bus, USB Interface, USB Connector, IrDA, CAN, Bluetooth, ISA, Parallel Peripheral Interfaces, PCI, PCI-X

FPGA: Field Programmable Devices, Programmability of FPGA, FPGA Logic Block, FPGA Design Flow

Module –III (12 hrs)

Modelling Techniques: Software and programming concept: Processor selection for An embedded system, State chart, SDL, Petri-Nets, Unified Modeling Language (UML).Hardware software co-design. Hardware and software partitioning: K-L partitioning, Partitioning using genetic algorithm .**Low power embedded system design:** Dynamic power dissipation, Static power dissipation, Power reduction techniques.

Text Books:

1. "Embedded system architecture, programming and design" By Raj Kamal, TMH.
2. "Embedded System Design " by Santanu Chattopadhay, PHI

Reference Books:

1. "Hardware software co-design of Embedded systems" By Ralf Niemann, KulwerAcademic.
2. "Embedded real time system programming" By Sriram V Iyer, Pankaj Gupta,TMH

Abbreviations Used:L = Lectures, P = Practical or Laboratory, T = Tutorial

IA = Internal Assessment , PA = Practical Assessment, EA = End-Semester Assessment

INTRUSION DETECTION SYSTEMS

Prerequisites	Computer Networks, Firewall, Classification Techniques
Course outcome	• Understand fundamental concepts of Network Protocol Analysis • Demonstrate the skill to capture and analyze network packets. • Use various protocol analyzers and Network Intrusion Detection Systems

Module-I
History of Intrusion detection, Audit, Concept and definition , Internal and external threats to data, attacks, Need and types of IDS, Information sources Host based information sources, Network based information sources.

Module-II
Intrusion Prevention Systems, Network IDs protocol based IDs, Hybrid IDs, Analysis schemes, thinking about intrusion. A model for intrusion analysis, techniques Responses requirement of responses, types of responses mapping responses to policy Vulnerability analysis, credential analysis non credential analysis

Module-III
Introduction to Snort, Snort Installation Scenarios, Installing Snort, Running Snort on Multiple Network Interfaces, Snort Command Line Options. Step-By-Step Procedure to Compile and Install Snort Location of Snort Files, Snort Modes Snort Alert Modes

Module-IV
Working with Snort Rules, Rule Headers, Rule Options, The SnortConfiguration File etc. Plug-in, Preprocessors and Output Modules, Using Snort with MySQL , Agent development for intrusion detection, Architecture models of IDs and IPs.

Text Books:
Rafeeq Rehman: “Intrusion Detection with SNORT, Apache, MySQL, PHP and ACID,” Prentice Hall

Reference Books:
1. Christopher Kruegel,Fredrik Valeur, Giovanni Vigna: “Intrusion Detection and Correlation Challenges and Solutions”, 1st Edition, Springer
2. Carl Endorf, Eugene Schultz and Jim Mellander “Intrusion Detection & Prevention”, 1st Edition, Tata McGraw-Hill,
3. Stephen Northcutt, Judy Novak: “Network Intrusion Detection”, 3rdEdition, New Riders Publishing,
4. T. Fahringer, R. Prodan, “A Text book on Grid Application Development and Computing Environment”. 6th Edition, Khanna Publishers