MANIPUR UNIVERSITY
CANCHIPUR, IMPHAL

CURRICULUM & SYLLABUS

FOR

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

(SECOND YEAR TO FOURTH YEAR)

(Effective from the Academic Session 2021-2022)

C O N T E N T S

Vision of the Institute

Excellence in engineering and technology education with good leadership in Human Resource Development.

Mission of the Institute

1. To produce technically strong, innovative, research oriented, all round developed engineers capable to solve modern challenges by adopting student centric teaching learning methods.
2. To impart engineering and technology education for all round development
3. To produce good engineering professional with social commitment.

About the Department

The Department has several state of the art Computer Laboratories with a host of servers, workstation and a large number of i3 and i5 Desktop Computers connected to the Campus-wide LAN with access to the internet through NKN (National Knowledge Network). The system runs on wide variety of operating systems including Linux Red hat, Windows 7 Professional, Windows 8, High profile Anti-Virus, MySQL, Oracle, Visual Studio, Adobe Premier, Maya (Latest). The laboratories are equipped with up-to-date office automation software, file servers in addition to the various state-of-the-art compilers and programming environment i.e. Hardware Lab., Software Lab., Networking Lab., and Graphic Lab. well equipped with latest Core i3, i5 PC's etc. The network laboratory is equipped with a wireless networking system, LAN trainers and Wi-Fi connection. There is a hardware Lab. equipped with various training kits, experimental setup, analyser equipment etc.

The Department presently offers the following programmes:

1. B.E. in Computer Science & Engineering
2. M. Tech. in Computer Science & Engineering

Vision of the Department

To promote scientific and quality technical education in Computer Science and Engineering and thereby generating competent professionals.

Mission of the Department

1. To impart scientific skills, moral and ethical values to the students
2. To serve the needs of industry, government and society through continuous human resource development.
3. To impart knowledge through quality research in emerging areas in the field of computer science and engineering
4. To fill the gap by undertaking collaborative projects which offer opportunities for long term interaction with academia and industry

Programme Specific Objectives (PSO)

PSO 1: Apply the fundamentals of Computer science and engineering knowledge to solve real world problems.

PSO 2: Apply the appropriate techniques and modern engineering hardware and software tools in Computer Science and engineering to engage in life-long learning and to successfully adapt in multi-disciplinary environments.

PSO 3: Apply the contextual knowledge of Computer Science and Engineering to address professional, societal and environmental issues with work ethics and function effectively as a team member.

Programme Educational Objectives (PEOs)

PEO1: To become a successful computer professional in the IT industry and related areas.

PEO2: To enable students to pursue higher education so that they can keep themselves up-to-date research & development.

PEO3: Graduates are prepared to communicate effectively with team members, engage in applying technologies and lead teams in industry.

PEO4: To develop professional skills and ethics in students so that they can prepare themselves for immediate employment and for lifelong learning in computer science and engineering.

PEO5: To demonstrate to the students the tools and techniques of Computer Science and Engineering program so that they can create innovative products for the benefit of society

Programme Outcomes (POs)

PO1: Ability to apply computer science theory in designing of computer systems for the benefit of the society.

PO2: Ability to develop algorithms for real world computational problems and analyze their complexities.

PO3: Ability to maintain computing systems using mathematics, engineering and programme courses.

PO4: Ability to design and implement computer systems with assured quality and efficiency.

PO5: Ability to analyze and design solutions for complex engineering problems which caters to the specified needs.

PO6: Ability to analyze large data samples and discover knowledge to provide solutions to engineering problems.

PO7: Ability to assess security, privacy, quality and cost parameters in developing software systems.

PO8: Ability to communicate effectively the engineering solutions to customers.

PO9: Ability to work with team members using common tools and environment to achieve project objectives.

PO10: Ability to engage in lifelong learning for higher studies.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CURRICULUMS

SECOND YEAR
THIRD SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	MA 231	ENGINEERING MATHEMATICS–III	3	1	0	70	30	-	100	4
2.	MA 232	DISCRETE MATHEMATICS	3	0	0	70	30	-	100	3
3.	CS 231	OBJECT ORIENTED PROGRAMMING	3	0	0	70	30	-	100	3
4.	CS 232	DATA STRUCTURES	3	1	0	70	30	-	100	4
5.	CS 233	DIGITAL LOGIC DESIGN	3	0	0	70	30	-	100	3
6.	HU231	ECONOMICS FOR ENGINEERS	2	0	0	35	15	-	50	2
		SUB-TOTAL	**17**	**2**	**0**				**550**	**19**
		PRACTICAL/DESIGN								
7.	CS 231P	OBJECT ORIENTED PROGRAMMING LAB	0	0	2		15	35	50	1
8.	CS 232P	DATA STRUCTURES LAB	0	0	4	-	30	70	100	2
*9.	CS 234P	INTERNSHIP-I	-	-	-		100	-	100	1
		SUB-TOTAL							**250**	**4**
		Mandatory Course								
10.	NC 231	CONSTITUTION OF INDIA/ ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	3	0	0	-	*50			
		TOTAL	**20**	**2**	**6**				**800**	**23**

*Mark or grade will be evaluated based on the internship of previous semester

SECOND YEAR
FOURTH SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	MA 241	NUMERICAL METHODS AND COMPUTATION	3	0	0	70	30	-	100	3
2.	CS 241	COMPUTER ORGANIZATION AND ARCHITECTURE	3	0	0	70	30	-	100	3
3.	CS 242	SYSTEM PROGRAMMING	3	0	0	70	30	-	100	3
4.	CS 243	DESIGN AND ANALYSIS OF ALGORITHMS	3	0	0	70	30	-	100	3
5.	CS 244	FORMAL LANGUAGES AND AUTOMATA THEORY	3	0	0	70	30	-	100	3
6.	HU 241	UNIVERSAL HUMAN VALUES-II	2	1	0	70	30	-	100	3
		SUB-TOTAL	**17**	**1**	**0**				**600**	**18**
		PRACTICAL/DESIGN								
7.	CS 241P	COMPUTER ORGANIZATION AND ARCHITECTURE LAB.	0	0	3	-	15	35	50	1.5
8.	CS 242P	SYSTEM PROGRAMMING LAB.	0	0	4	-	30	70	100	2
9.	CS 243P	DESIGN AND ANALYSIS OF ALGORITHMS LAB.	0	0	4	-	30	70	100	2
		SUB-TOTAL							**250**	**5.5**
		MANDATORY COURSE								
10.	NC 241	ENVIRONMENTAL SCIENCES	2	0	0	-	*50			
		TOTAL	**19**	**1**	**11**				**850**	**23.5**

THIRD YEAR
FIFTH SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	CS 351	OPERATING SYSTEM	3	0	0	70	30	-	100	3
2.	CS 352	DATABASE MANAGEMENT SYSTEM	3	0	0	70	30	-	100	3
3.	CS 353	GRAPH THEORY	3	0	0	70	30	-	100	3
4.	CS 354	COMPUTER NETWORKS	3	0	0	70	30	-	100	3
5.	CS 355	ARTIFICIAL INTELLIGENCE	3	0	0	70	30	-	100	3
6.	CS 356	PROGRAM ELECTIVE – I	3	0	0	70	30	-	100	3
		SUB-TOTAL	**18**	**0**	**0**				**600**	**18**
		PRACTICAL/DESIGN								
7.	CS 352P	DATABASE MANAGEMENT SYSTEM LAB.	0	0	4	-	30	70	100	2
8.	CS 356P	PROGRAM ELECTIVE – I LAB.	0	0	4	-	30	70	100	2
*9.	CS 357P	INTERNSHIP - II	-	-	-	-	100	-	100	1
		SUB-TOTAL	**0**	**0**	**8**				**300**	**5**
		MANDATORY COURSE								
10	**NC 351**	**ORGANIZATIONAL BEHAVIORS**	3	0	0	-	*50	-		
		TOTAL	**21**	**0**	**8**				**900**	**23**

*Mark or grade will be evaluated based on the internship of previous semester

THIRD YEAR
SIXTH SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	CS 361	COMPILER DESIGN	3	0	0	70	30	-	100	3
2.	CS 362	SOFTWARE ENGINEERING	3	0	0	70	30	-	100	3
3.	CS 363	PROGRAM ELECTIVE – II	3	0	0	70	30	-	100	3
4.	CS 364	PROGRAM ELECTIVE – III	3	0	0	70	30	-	100	3
5.	CS 365	OPEN ELECTIVE – I	3	0	0	70	30	-	100	3
		SUB-TOTAL							**500**	**15**
		PRACTICAL/DESIGN								
6.	CS 361P	COMPILER DESIGN LAB.	0	0	4	-	30	70	100	2
7.	CS 362P	SOFTWARE ENGG. LAB.	0	0	4	-	30	70	100	2
8.	CS 366P	MINI PROJECT	0	0	8	-	60	140	200	3
		SUB-TOTAL							**400**	**7**
		TOTAL	**15**	**0**	**16**				**900**	**22**

FOURTH YEAR
SEVENTH SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	CS 471	PROGRAM ELECTIVE –IV	3	0	0	70	30	-	100	3
2.	CS 472	PROGRAM ELECTIVE –V	3	0	0	70	30	-	100	3
3.	CS 473	PROGRAM ELECTIVE –VI	3	0	0	70	30	-	100	3
4.	CS 474	OPEN ELECTIVE – II	3	0	0	70	30	-	100	3
		SUB-TOTAL	**12**	**0**	**0**				**400**	**12**
		PRACTICAL/DESIGN								
5.	CS 475P	PROJECT-I(PROJECT WORK, SEMINAR AND INTERNSHIP IN INDUSTRY OR AT APPROPRIATE WORK PLACE)	0	0	12	-	90	210	300	3
6.	CS 476P	INTERNSHIP-III	0	0	0	-	100	-	100	1
		SUB-TOTAL							**400**	**4**
		TOTAL	**12**	**0**	**12**				**800**	**16**

*Mark or grade will be evaluated based on the internship of previous semester

FOURTH YEAR
EIGHTH SEMESTER

Sl. No.	Course No.	Subject	Hours/Week			Marks				Credit
			L	T	P	Theory	Sess.	Pract.	Total	
1.	CS 481	PROGRAM ELECTIVE – VII	3	0	0	70	30	-	100	3
2.	CS 482	PROGRAM ELECTIVE – VIII	3	0	0	70	30	-	100	3
3.	CS 483	OPEN ELECTIVE-III	3	0	0	70	30	-	100	3
		SUB-TOTAL	**9**	**0**	**0**				**300**	**9**
		PRACTICAL/DESIGN								
4.	CS 484P	PROJECT-II (CONTINUED FROM VII SEMESTER, PROJECT WORK, SEMINAR AND INTERNSHIP IN INDUSTRY OR AT APPROPRIATE WORK PLACE)	0	0	16	-	120	280	400	8
		SUB-TOTAL	**0**	**0**	**16**				**400**	**8**
		TOTAL	**9**	**0**	**16**				**700**	**17**

DETAILED SYLLABUS (THIRD SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
MA 231	ENGINEERING MATHEMATICS–III	3	1	0	70	30	-	100	4

Course Objectives:

1. To understand Derichlet's condition, General Fourier series, complex form of Fourier series.
2. To understand F. S. of odd and even functions, F. S. of half range.
3. To understand Fourier transform and Determination of Fourier sine, cosine transforms.
4. To understand second order quasi linear PDE's and Method of separation of variables.
5. To solve one dimensional wave and heat equations, two dimensional heat equation.
6. To understand analytical functions.
7. Apply to contour integration over circle and semi-circle excluding poles on real axis.

Module 1: Fourier Series (6 lectures)

Derichlet‟s condition-General Fourier series- odd and even functions, Half range-sine and cosine series-complex form of Fourier series, Practical Harmonic analysis.

Module 2: Boundary Value Problems (6 lectures)

Classification of second order quasi linear partial differential equations- solution of one dimensional wave equation, one dimensional heat equation- steady state solution of two dimensional heat equation(insulated edges exclude)-Solution by separation of variables.

Module 3: Complex Analysis (10 lectures)

Analytic function-properties, Cauchy-Riemann equations, construction of analytic function, determination of conjugate harmonic functions, application to two dimensional potential problems. Conformal transformations, Bilinear Transformation. Cauchy's Integral theorem and Cauchy's integral formula (statement only), Taylor's and Laurent's expansions, isolated singularities, residues-Cauchy's residues theorem (statement only), contour integration-over unit circle and semi- circle(excluding poles on real axis).

Module 4: Basic Probability (10 lectures)

Probability spaces, conditional probability, independence; Discrete random variables, Independent random variables, the multinomial distribution, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality.

Module 5: Basic Statistics (8 lectures)

Measures of Central tendency: Moments, skewness and Kurtosis - Probability distributions: Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions, Correlation and regression – Rank correlation.

Suggested Text & Reference Books:

1. B. S. Grewal, Higher Engineering Mathematics, Khanna Publication, 41 Edition, New Delhi.
2. Erwin Kreyszig, Advance Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
3. Ramana B. V., Higher Engineering Mathematics, Tata McGraw Hill, New Delhi, 11th Reprint, 2010.
4. N. O. Bali and Manish Goyal, A Text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
5. P. G. Hoel, S. C. Port and C. J. Stone, "Introduction to Probability Theory", Universal Book Stall, 2003.
6. S. Ross, "A First Course in Probability", Pearson Education India, 2002.
7. W. Feller, "An Introduction to Probability Theory and its Applications", Wiley, 1968.

Course Outcomes:
The objective of this course is to familiarize with basics of Fourier series, Boundary value problems, calculus of complex variable, basics of probability and statistics. The students will learn:

- The methods to expand a function in Fourier series.
- The methods to solve partial differential equations that are arising in engineering Problems.
- The tools of differentiation and integration of functions of a complex variable that are used in various techniques dealing with engineering problems.
- The basics of probability and statistics that are essential in most branches of engineering

.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
MA 232	DISCRETE MATHEMATICS	3	0	0	70	30	-	100	3

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:

1. Use mathematically correct terminology and notation.
2. Construct correct direct and indirect proofs.
3. Use division into cases in a proof.
4. Use counter examples.
5. Apply logical reasoning to solve a variety of problems.

Module1:Logic:
Statements – Truth Tables – connectives-Normal forms-Predicate Calculus- Inference theory for statement calculus and Predicate Calculus.

Module2:Combinatorics:
Review of Permutation and combination-Mathematical Induction-Pigeon hole principle – Principle of inclusion and exclusion-Generating function- Recurrence relations

Module3:Groups:
Semi groups – Monoids – groups – permutation group – Concepts – Language's theorem – Group homomorphism-Kernel, Rings and Fields (definitions and simple examples only)

Module4:Lattices:
Partial ordering-Posets-Hass Lattices-Properties of Lattices-Sub Lattices- Special Lattices-Boolean Algebra.

Module5:Graphs:
Introduction to Graphs-Graph terminology – Representation of Graphs – Graph Isomorphism-Connectivity-Euler and Hamilton Paths.

Suggested Books:

1. "Discrete Mathematical Structure with Applications to Computer Science" by Tremlay J.P. and Manohar R. McGraw Hill Book Company
2. "Discrete and Combinatorial mathematics", Ralph P. Grimaldi Áddision Wesley, Publishing Company: Reprinted in 1985 Section : 1:1
3. "Discrete Mathematics and Its Application", Kenneth H. Rosen McGraw Hill Book Company

Course Outcomes

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

1. Express a given logic sentence in terms of predicates, quantifiers and logical connectives
2. Derive the solution using deductive logic and prove the solution based on logical inference, for a given a problem
3. Classify algebraic structure for a given mathematical problem
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5. Develop the given problem as graph networks and solve them with techniques of graph theory.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 231	OBJECT ORIENTED PROGRAMMING	3	0	0	70	30	-	100	3

Course Objectives

The objective of course is to develop programming skills of students, using object oriented programming concepts, learn the concept of class and object using C++ and develop classes for simple applications.

1. Identify the importance of object oriented programming and the difference between structured oriented and object oriented programming features.
2. Able to make use of objects and classes for developing programs.
3. Able to use various object oriented concepts to solve different problems.

Course contents:

Module1: Introduction: Overview of C++, Sample C++ program, Different data types, operators, expressions, and statements, arrays and strings, pointers & user defined types Function Components, argument passing, inline functions, function overloading, recursive functions.

Module2: Classes & Objects – I: Class Specification, Class Objects, Scope resolution operator, Access members, Defining member functions, Data hiding, Constructors, Destructors, Parameterized constructors, Static data members, Functions- Friend functions, Passing objects as arguments, Returning Objects, Arrays of objects, Dynamic objects, Pointers to objects, Copy constructors, Generic functions and classes, Applications Operator Overloading using friend functions such as +, - , pre-increment, post-increment,[ ] etc., overloading <>.

Module3: Inheritance – Base Class, Inheritance and protected members, Protected Base class inheritance, Inheriting multiple base classes - Constructors, Destructors and Inheritance, Passing parameters to base class constructors, Granting access, Virtual base classes

Module4: Virtual functions, Polymorphism: Virtual function, Calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions and hierarchical, Pure virtual functions, Abstract classes, Using virtual functions, Early and late binding.

Module5: I/O System Basics, File I/0: C++ stream classes, Formatted I/O, I/O manipulators, fstream and the File classes, File operations

Suggested Text Book & References:

1. Balagurusamy , Object Oriented Programming with C++, TMH
2. Robert Lafore, Object-Oriented Programming in C++, Third Edition, Macmillan Computer Publishing
3. Rajesh Shukla, Object Oriented Programming With C++ Wiley India.
4. Rohit Khurana, Object Oriented Programming With C++ , Vikas Publishing House
5. Sourav Sahay, Object Oriented Programming with C++, Oxford University Press.

Course Outcomes

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

1. Distinguish between Structured and Object Oriented problem solving approaches and apply them based on the problem given
2. Identify classes and objects from the given problem description and able to create classes and objects using C++
3. Improve secured data processing by applying Abstraction, Encapsulation and Information hiding
4. Achieve code reusability and extensibility by means of Inheritance and Polymorphism.
5. Handle Exceptions and identify Run Time Type Information (RTTI)
6. Translate the informal description of an algorithm to solutions for problems in engineering, science and text processing using Object Oriented Programming.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 232	DATA STRUCTURES	3	1	0	70	30	-	100	4

Course Objectives:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures

Course contents:

Module1:

Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc. Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Linear Search and Binary Search Techniques and their complexity analysis.

Module2:

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation–corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each type of Queues: Algorithms and their analysis.

Module3:

Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

Trees: Basic Tree Terminologies, Different types of Trees , Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. BTree, B+Tree: definitions, algorithms and analysis.

Module4:

Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison Among All The Methods, Hashing.

Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

Suggested Books:

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

Suggested books:

1. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer ", R.G.Dromey, Pearson Education.

Course Outcomes

1. For a given algorithm student will be able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search) students will be able to implement it.
3. For a given problem of Stacks, Queues and linked list students will be able to implement it and analyze the same to determine the time and computation complexity.
4. Students will be able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in terms of Space and Time complexity.
5. Students will be able to implement Graph search and traversal algorithms and determine the time and computation complexity.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 233	DIGITAL LOGIC DESIGN	3	0	0	70	30	-	100	3

Course Objectives:

1. To learn number systems.
2. To learn Boolean functions.
3. To minimize Boolean functions.
4. To design and analyze combinational logic circuits.
5. To understand the basic software tools for the design and implementation of digital circuits and systems.

Binary Systems:

Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic.

Boolean Algebra and LOGIC GATES:
Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, other logic operations, Digital logic gates, integrated circuits.

GATE – Level Minimization:
The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementations, Exclusive – OR function

Combinational LOGIC :
Combinational Circuits, Analysis procedure Design procedure, Binary Adder- Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers

Synchronous Sequential LOGIC:
Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, State Reduction and Assignment, Design Procedure.

Registers and Counters:
Registers, shift Registers, Ripple counters, Synchronous counters, other counters.

Memory and Programmable LOGIC:
Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Suggested Text Book & References:

1. M. Moris Mano, Charles R. Kime, Logic and Computer Design Fundamentals, 2nd Edition, PEARSON EDUCATION Asia
2. M. Moris Mano, Michael D. Ciletti, Digital Design, Prentice HALL.
3. Stephen Brown Zvonko Vranesic, Fundamentals of Digital Logic with VHDL design, McGraw Hill
4. Enocho Hwang, Digital Logic and Microprocessor Design with VHDL, Thomson,
5. John F. Wakerly, Digital design Principles & Practices, 3rd EDITION, Prentice HALL.
6. H.T. Nagle, Introduction to Computer Logic, Prentice Hall.

Course Outcome:
On the successful completion of the course, students will be able to
1. Understand the fundamental concepts and techniques used in digital electronics.
2. Understand various number systems and its application in digital design.
3. Understand various combinational and sequential circuits.
4. Design various combinational and sequential circuits.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
HU231	ECONOMICS FOR ENGINEERS	2	0	0	35	15	-	50	2

Course Objectives:
1. To explain engineering economics' decision making process, state and explain the law of demand and supply, law of diminishing returns and solve problems on interest factors.

2. To understand the relevance of present worth and future worth comparisons and compare the Net present worth and Net future worth of co-terminated assets, assets with unequal lives and infinite lives.
3. To understand the relevance of Equivalent annual worth comparisons and compare the equivalent annual worth of co-terminated assets and assets with unequal lives and infinite lives.
4. To appraise investments using non-discounted and discounted cash flow techniques, show the depreciation calculations using straight line and declining balance methods and explain the procedure for determining the corporate income tax.
5. To explain the various cost concepts, cost estimation methods, the principles of cost accounting, the elements of balance sheet and profit and loss account and the use of financial ratios for measuring financial performance.

Course Content

UNIT - 1

Introduction: Engineering Decision-Making, Engineering and Economics, Problem solving and Decision making, Intuition and Analysis, Tactics and Strategy. Engineering Economic Decision, Maze. Law of demand and supply, Law of returns, Interest and Interest factors: Interest rate, Simple interest, Compound interest, Cash - flow diagrams, Personal loans and EMI Payment, Exercises and Discussion. **10 hours**

UNIT – 2

Present-Worth and Future worth Comparisons: Conditions for present worth comparisons, Basic Present worth comparisons, Present-worth equivalence, Net Present-worth, Assets with unequal lives, infinite lives, Future-worth comparison, Pay-back comparison, Exercises, Discussions and problems. **10 Hours**

UNIT– 3

Equivalent Annual-Worth Comparisons: Equivalent Annual-Worth Comparison Methods, Situations for Equivalent Annual-Worth Comparisons, Consideration Of asset life, Comparison Of assets with equal and unequal lives, Use Of Sinking Fund Method, Annuity Contract For Guaranteed income, Exercises, Problems. **12 Hours**

UNIT– 4

Rate-Of-Return Calculation and Depreciation: Rate of return, Minimum acceptable rate of return, IRR, IRR misconceptions, Cost Of Capital concepts. Causes of Depreciation, Basic Methods Of computing depreciation charges, Tax Concepts, corporate income tax. **10 hours**

UNIT– 5

Introduction to land Scope of Finance Functions: Statements of Financial Information: Introduction, Source Of Financial Information, financial statements, Balance Sheet, and Profitand Loss Account, relation between Balance sheet and Profit and Loss account. Financial performance analysis using financial ratios, Simple Numerical. **10 hours**

Text Books:

1. James L Riggs, Engineering Economy, McGraw Hill, 2002.

Reference Books:

1. Gerald J Thuesen, Engineering economy, Prentice-Hall-India, Pvt Ltd, 2002.
2. Prasanna Chandra, Financial Management, Tata McGraw Hill, 200

Course Outcomes:

1. Evaluate the economic theories ,cost concepts and pricing policies
2. Understand the market structures and integration process.
3. Understand the measures of national income, the function of banks and concepts of globalization.
4. Apply the concepts of financial management for project appraisal.
5. Understand accounting system.
6. Understand the impact of inflation, taxation and depreciation.

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 231P	OBJECT ORIENTED PROGRAMMING LAB.	0	0	2		15	35	50	1

Laboratory practical based on CS 231.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 232P	DATA STRUCTURES LAB	0	0	4	-	30	70	100	2

Laboratory practical based on CS 232.

MANDATORY COURSE

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
NC 231	CONSTITUTION OF INDIA/ ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	3	0	0	-	*50		50	0

Constitution of India - Basic features and fundamental principles: The Constitution of India is the supreme law of India. Parliament of India can not make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the "basic structure" of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments.

The Constitution of India Reflects the idea of "Constitutionalism" – a modern and progressive concept historically developed by the thinkers of "liberalism" – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of "constitutionalism" in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India's legacy of "diversity". It has been said that Indian constitution reflects ideals of its freedom movement, however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be "static" and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it "as one of the strongest court in the world".

Course content

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India
4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India
9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
11. Emergency Provisions : National Emergency, President Rule, Financial Emergency
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article 19
15. Scope of the Right to Life and Personal Liberty under Article 2

DETAILED SYLLABUS (FOURTH SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
MA 241	NUMERICAL METHODS AND COMPUTATION	3	0	0	70	30	-	100	3

Course Objectives:
The main objective of this course is:
1. Introduction to Computer arithmetic and errors.
2. Introduction to solution of Algebraic and Transcendental equation, Linear System of equations and differential equations
3. Introduction to Eigenvalues and Eigenvectors.

Unit-1: Accuracy and Errors in Computation:
Basic concepts on accuracy of numbers, significant figures, rounding of numbers, Errors representation-Inherent and truncation, Absolute and relative errors, errors in the approximation and series approximation of functions.

Unit-2: Solution of Algebraic and Transcendental Equations:
Bisection, Regula-Falsi method, Newton-Raphson and iterative methods with their convergence conditions, Generalizations of Newton-Raphson and iterative methods with simultaneous nonlinear equations.

Unit-3: Solution of Linear System of Equations:
Gaussian Elimination Method With Partial Pivoting, Factorization Method, Matrix Inverse method, Gauss-Jacobi and Gauss-Seidel iterative methods, Fitting of curve by method of curve by method of least square.

Unit-4:Numerical solution of Differential Equations:
Picard's and Taylor series, Euler's Method And its modified form, Runge-Kutta methods, Solution of two-point boundary value problems using finite difference: One dimensional parabolic equations-Schmidt method, Crank-Nicolson method; Elliptic equations-Solution of Laplace equation-Jacobi and Gauss-Seidel methods; Hyperbolic Equations: Solution Wave Equation.

Unit-5:Eigenvalues and Eigen-vectors problems:
Gershgorin's Theorem (without proof) with simple problems, Power method for dominant Eigen-values, Jacobian method for symmetric matrices.

Suggested Text Books & References:

1. Grewal B.S., "Numerical Methods", Khanna Pub., New Delhi
2. Shartry S.S., "Numerical Methods", Prentice Hall Inc., India
3. C.F. Gerald and P.O. Wheatley, "Applied Numerical Analysis", Addison Wesley,
4. J.H. Wilkinson, "Algebraic Eigen –Value Problems", Oxford Univ. Press
5. G.D. Smith, "Numerical Solution of Partial Differential Equations", Oxford Univ. Press

Course Outcomes:
Students completing the course will be able to
- Understand fundamental arithmetic operations and Errors Estimation.
- Solve algebraic and transcendental equations.
- Solve the system of linear equations of various methods.
- Solve the differential equations.
- Determine the Eigenvalues and Eigenvectors.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 241	COMPUTER ORGANIZATION AND ARCHITECTURE	3	0	0	70	30	-	100	3

Course Objectives:

1. To understand the design of the various functional units and components of computers.
2. To identify the elements of modern instruction sets and their impact on processor design.
3. To explain the function of each element of a memory hierarchy,
4. To impart the knowledge of microprogramming
5. Understand the concepts of pipelining techniques.

Detailed contents:

Module 1:

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs.

Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and add, Booth multiplier. Division restoring and non-restoring techniques, floating point arithmetic.

Module 2:

Introduction to x86 architecture.

CPU control unit design: hardwired and microprogrammed design approaches, Case study – design of a simple hypothetical CPU.

Memory system design: semiconductor memory technologies, memory organization.

Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB

Module 3:

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.

Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

Module 4:

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Suggested books:

1. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
2. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

Suggested Reference books:

1. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
3. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course Outcomes:

1. Understand the theory and architecture of the central processing unit.
2. Analyze some of the design issues in terms of speed, technology, cost, performance.
3. Design a simple CPU with applying the theory concepts.
4. Use appropriate tools to design, verify and test the CPU architecture.
5. Learn the concepts of parallel processing, pipelining and interprocessor communication.
6. Understand the architecture and functionality of the central processing unit.
7. Exemplify in a better way the I/O and memory organization.
8. Define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 242	SYSTEM PROGRAMMING	3	0	0	70	30	-	100	3

Course Objectives:

1. Write interactive shell scripts.
2. Compile a C program on Linux using cc/gcc command.
3. Automate the creation of programs using the make utility in Linux.
4. Integrate C programs with Shell scripts.
5. Appreciate the shell interface with awk.
6. Enumerate different ways to define the structure of an input file in awk.
7. Use awk for text processing

Detailed Contents:

Unit-1: Linux commands and Utilities-File and Directory Operations-Linux Shells-Types of Shells, Writing & Executing Basic Script, Debugging Script, Making Interactive Scripts, Variables, Mathematical Expressions- Conditional Statements- Logical Operators-Loops-Positional Parameters, set & shift, IFS, break & continue- File Manipulations.

Unit-2: Linux System Calls-File Handling using System Calls-Process Control System Calls- Write, Compile, and Execute C programs- make Utility- Integrate C programs with Shell Scripts.

Unit-3: awk Background- Features of awk- Invoking awk- Fields and Records-Patterns in awk- Regular Expressions- Relational Operators- Print Redirection- Variables- Arrays-Flow controls- The shell and awk.

Suggested Text Books:

1. Linux: The Complete Reference, by Richard Petersen, McGraw-Hill Education, 2007.
2. C Programming Language, Kernighan and Ritchie, Pearson, 1988
3. Linux Command Line and Shell Scripting, Richard Blum and Christine Bresnahan, Wiley, 2021

Course Outcomes:

Students will be able to

1. Describe and apply the basic set of commands and utilities in Linux systems.
2. Write shell scripts to perform repetitive tasks.
3. Automate the creation of programs using the make utility in Linux.
4. Integrate C programs with Shell scripts.
5. Use awk scripts to design programs for users.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 243	DESIGN AND ANALYSIS OF ALGORITHMS	3	0	0	70	30	-	100	3

Course Objectives:

The objectives of this course are to

1. Introduce students to fundamental data structures; their algorithms, implementations and applications.
2. Teach students how to analyze the efficiency of the fundamental data structures in terms of both time and space so that they are able to decide what data structure is suitable for a given problem.
3. To explain different computational models (e.g., divide-and-conquer), order notation and various complexity measures (e.g., running time, disk space) to analyze the complexity/performance of different algorithms.

Module 1:

Introduction: Euclid's algorithm Problem, Instance, RAM model, Asymptotic complexity of Algorithms

Module 2:

Divide and conquer: Introduction, Binary Sort, quick sort, Strassen multiplication, Median Finding

Module 3:

Dynamic Programming: Shortest Path in graph, Knapsack, Longest common subsequence, Matrix chain multiplication or Optimal search trees, TSP, A machine scheduling problem.

Module 4:

Greedy Algorithms: Introduction, Set of Intervals, Fractional Knapsack, Minimum cost spanning tree, Huffman Coding, single source shortest path.

Module 5:

Back Tracking: 8-queen problem, knapsack problem

Module 6:

Branch and bound: TSP

Suggested Text Book & References:

1. S. K. Basu, Design Methods and Analysis of Algorithms, PHI
2. Parag Himanshu Dave, Design and Analysis of Algorithms, Pearson Education India
3. Chandra Mohan, Design and Analysis of Algorithms, Prentice-Hall India Pvt. Limited.
4. Cormen, Thomas, Introduction to Algorithms, PHI, New Delhi

Course Outcomes:

After completion of this course, the student shall be able to:

1. Analyze the asymptotic performance of algorithms.
2. Implement various data structures and their algorithms, and apply them in implementing simple applications.
3. To analyze simple algorithms and determine their efficiency using big -O notation.
4. Apply important algorithmic design paradigms and methods of analysis.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 244	FORMAL LANGUAGES AND AUTOMATA THEORY	3	0	0	70	30	-	100	3

Course Objectives:

1. Design finite automata to accept a set of strings of a language.
2. Prove that a given language is regular and apply the closure properties of languages.
3. Design context free grammars to generate strings from a context free language.
4. Prove equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
5. Identify the hierarchy of formal languages, grammars and machines.
6. Design of Turing machines

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages.

Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Turing Machine : Turing Machine, design of TMs, Computable functions, Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties.

References:

1. Mishra and Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI
2. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Course Outcomes:

1. Design finite automata to accept a set of strings of a language.
2. For a given language, determine whether the given language is regular or not.

3. Design context free grammars to generate strings of context free language.
4. Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
5. Write the hierarchy of formal languages, grammars and machines.
6. Design of a Turing for a given function.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
HU 241	UNIVERSAL HUMAN VALUES-II	2	1	0	70	30	-	100	3

1. OBJECTIVE:

The objective of the course is four fold:

1. Development of a holistic perspective based on self-exploration about themselves (human being), family, society and nature/existence.
2. Understanding (or developing clarity) of the harmony in the human being, family, society and nature/existence
3. Strengthening of self-reflection.
4. Development of commitment and courage to act.

2. COURSE TOPICS:

The course has 28 lectures and 14 practice sessions in 5 modules:

Module 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education

1. Purpose and motivation for the course, recapitulation from Universal Human Values-I
2. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and Experiential Validation-as the process for self-exploration
3. Continuous Happiness and Prosperity-A look at basic Human Aspirations
4. Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority
5. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario
6. Method to fulfill the above human aspirations: understanding and living in harmony at various levels.

Include practice sessions to discuss natural acceptance in human being as the innate acceptance for living with responsibility (living in relationship, harmony and co-existence) rather than as arbitrariness in choice based on liking-disliking

Module2: Understanding Harmony in the Human Being-Harmony in Myself!

1. Understanding human being as a co-existence of the sentient 'I' and the material 'Body'
2. Understanding the needs of Self ('I') and 'Body' - happiness and physical facility
3. Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer)
4. Understanding the characteristics and activities of 'I' and harmony in 'I'
5. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail
6. Programs to ensure Sanyam and Health.

Include practice sessions to discuss the role others have played in making material goods available to me. Identifying from one's own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs dealing with disease.

Module 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship

1. Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship
2. Understanding the meaning of Trust; Difference between intention and competence
3. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship
4. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals
5. Visualizing a universal harmonious order in society- Undivided Society, Universal Order-from family to world family.

Include practice sessions to reflect on relationships in family, hostel and institute as extended family, real life examples, teacher-student relationship, goal of education etc. Gratitude as a universal value in relationships. Discuss with scenarios. Elicit examples from students' lives.

Module 4: Understanding Harmony in the Nature and Existence - Whole existence as Coexistence.

1. Understanding the harmony in the Nature.
2. Interconnectedness and mutual fulfilment among the four orders of nature-recyclability and self regulation in nature
3. Understanding Existence as Co-existence of mutually interacting units in all-pervasive space
4. Holistic perception of harmony at all levels of existence.

Include practice sessions to discuss human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

Module 5: Implications of the above Holistic Understanding of Harmony on Professional Ethics

1. Natural acceptance of human values
2. Definitiveness of Ethical Human Conduct
3. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order
4. Competence in professional ethics:
    a. Ability to utilize the professional competence for augmenting universal human order
    b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems,
    c. Ability to identify and develop appropriate technologies and management patterns for above production systems.
5. Case studies of typical holistic technologies, management models and production systems.
6. Strategy for transition from the present state to Universal Human Order:
    a. At the level of individual: as socially and ecologically responsible engineers, technologists and managers
    b. At the level of society: as mutually enriching institutions and organizations
7. Sum-up.

Include practice Exercises and Case Studies will be taken up in Practice (tutorial) Sessions e.g. To discuss the conduct as an engineer or scientist etc.

3. READINGS:

Text Book

1. Human Values and Professional Ethics by RR Gaur, R Sangal, GP Bagaria, Excel Books, New Delhi, 2010

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth-by Mohandas Karamchand Gandhi
5. Small is Beautiful-E.F Schumacher.
6. Slow is Beautiful-Cecile Andrews
7. Economy of Permanence-JC Kumarappa
8. Bharat Mein Angreji Raj-Pandit Sunderlal
9. Rediscovering India-by Dharampal
10. Hind Swaraj or Indian Home Rule –by Mohandas K. Gandhi
11. India Wins Freedom- Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi-Romain Rolland (English)

4. MODE OF CONDUCT (2L:1T:0P) (3credits)

Lectures hours are to be used for interactive discussion, placing the proposals about the topics at hand and motivating students to reflect, explore and verify them.
Tutorial hours are to be used for practice sessions.

While analysing and discussing the topic, the faculty mentor's role is in pointing to essential elements to help in sorting them out from the surface elements. In other words, help the students explore the important or critical elements.

In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with one's own self and do self-observation, self-reflection and self-exploration.

Scenarios may be used to initiate discussion. The student is encouraged to take up "ordinary" situations rather than" extra-ordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentor,in a group sitting.

Tutorials (experiments or practical) are important for the course. The difference is that the laboratory is everyday life, and practical are how you behave and working real life. Depending on the nature of topics, worksheets, home assignment and/or activity are included.

The practice sessions (tutorials) would also provide support to a student in performing actions commensurate to his/her beliefs. It is intended that this would lead to development of commitment, namely behaving and working based on basic human values.

It is recommended that this content be placed before the student as it is, in the form of a basic foundation course, without including anything else or excluding any part of this content. Additional content may be offered in separate, higher courses.

This course is to be taught by faculty from every teaching department, including HSS faculty.

Teacher preparation with a minimum exposure to at least one 8-day FDP on Universal Human Values is deemed essential.

5. ASSESSMENT:

This is a compulsory credit course. The assessment is to provide a fair state of development of the student, so participation in classroom discussions, self-assessment, peer assessment etc. will be used in evaluation.

Example:

Assessment by faculty mentor	: 10 marks
Self-assessment	: 10marks
Assessment by peers	: 10marks
Socially relevant project/Group Activities/Assignments	: 20 marks
Semester End Examination	: 50marks

The overall pass percentage is 40%.
In case the student fails, he/she must repeat the course.

6. OUTCOME OF THE COURSE:

By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.

They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

This is only an introductory foundational input. It would be desirable to follow it up by

a) faculty-student or mentor-mentee programs throughout their time with the institution
b) Higher level courses on human values in every aspect of living. e.g. as a professional

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 241P	COMPUTER ORGANIZATION AND ARCHITECTURE LAB.	0	0	3	-	15	35	50	1.5

Laboratory practical based on CS 241.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 242P	SYSTEM PROGRAMMING LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 242.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 243P	DESIGN AND ANALYSIS OF ALGORITHMS LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 243.

MANDATORY COURSE

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
NC 241	ENVIRONMENTAL SCIENCES	2	0	0	-	*50			

MODULE 1: Environment and Ecology

1. Introduction
2. Environment and Ecology
3. Objectives of ecological study
4. Aspects of Ecology
   a) Autecology
   b) Synecology
5. Ecosystem
   a) Structural and functional attributes of an ecosystem
   b) Food chain and food web
   c) Energy flow
   d) Biogeochemical cycles

MODULE 2: Land: Use and Abuse

1. Land use: Impact of land – use on environmental quality
2. Land degradation
3. Control of land degradation
4. Waste land
5. Wet lands

MODULE 3: Water Pollution

1. Introduction
2. Water quality standards
3. Water pollution
4. Control of water pollution
5. Water pollution legislations
6. Water quality management in Rivers

MODULE 4: Air Pollution

1. Introduction
   a) Air pollution system
   b) Air pollutants
2. Air pollution laws
3. Control of air pollution
   a) Source correction method
   b) Pollution control equipment

MODULE 5: Noise Pollution

1. Introduction
2. Sources of noise pollution
3. Effects of noise
   a) Physical effects
   b) Physiological effects
   c) Psychological effects
4. Controls of Noise pollution

Text / Reference Books:

1. Environmental engineering and management by Dr Suresh Dhameja
2. Environmental studies by Dr B.S. Chauhan
3. Environmental science and engineering by Henry and Hence
4. Environmental studies for undergraduate course by Dr Susmitha Baskar
5. Chemistry for environmental engineering and science by Clair Sawyer

DETAILED SYLLABUS (FIFTH SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 351	OPERATING SYSTEM	3	0	0	70	30	-	100	3

Course Objectives

To learn the fundamentals of Operating Systems.

1. To learn the mechanisms of OS to handle processes and threads and their communication
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that include architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
4. To know the components and management aspects of concurrency management.

DETAILED CONTENTS :

Module 1:

Introduction: Concept of Operating Systems, Generation of Operating Systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept Virtual Machine. Case study on UNIX and WINDOWS Operating System.

Module 2:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block(PCB),Context Switching

Thread: Definition, Various States, Benefits Of Threads, Types Of Threads, Concept of Multi Thread.,

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling Criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time;

Scheduling Algorithms: Preemptive Non preemptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

Module 3:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem.

Module 4:

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's Algorithm, Deadlock detection and Recovery.

Module 5:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation–Fixed And Variable Partition– Internal and External fragmentation and Compaction; Paging: Principle of operation–Page Allocation– Hardware Support For Paging, Protection And Sharing, Disadvantages of Paging.

Virtual Memory: Basics of Virtual Memory–Hardware And control structures–Locality of Reference, Page Fault, Working Set, Dirty Page/Dirty Bit–Demand Paging, Page Replacement algorithms: Optimal, First in First Out(FIFO),Second Chance(SC),Not Recently Used(NRU) and Least Recently used(LRU).

Module 6:

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Secondary-Storage Structure: Disk Structure, Disk Scheduling Algorithms

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Disk Management: Disk Structure, Disk scheduling- FCFS, SSTF, SCAN, C-SCAN, Disk Reliability, Disk Formatting, Boot-block, Bad blocks

Suggested Books:

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested Reference Books:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Course Outcomes

1. Create Processes And Threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
3. For a Given Specification Of Memory Organization Develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and Implement File Management System.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 352	DATABASE MANAGEMENT SYSTEM	3	0	0	70	30	-	100	3

Course Objectives:

1. This course will give an opportunity to gain an idea on how data are actually stored and how its being accessed by the user.
2. It will provide you the concept of normalizing table, idea to build an Entity Relationship Diagram
3. It provides the assurance of security and integrity.

DETAIL CONTENTS :

Module 1: Introduction to Databases and Transactions.

Database system and applications, Purpose of database system, Database System architecture- levels, View of data, Mappings, Database, users and DBA

Module 2: Data Models, Database Design, ER-Diagram and Constraints

The importance of data models, Database design, ER-Model, Entity Set in DBMS, Relationship Sets, Cardinality in ER Diagram, Participation Constraints, Types of attribute, Converting ER Diagram to Tables, Problems on Converting ER Diagrams to Tables ER-Diagrams, ERD Issues, Constraints in DBMS, Handling Violation Referential Integrity Constraints.

Module 3: SQL

Basics of SQL, DDL,DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. transaction control commands –Commit, Rollback, Save point.

Module 4: Functional Dependency, Decomposition, Normalization , Transaction.

Concept of Functional Dependency, Functional Dependencies Equivalence, Canonical Cover of Functional Dependency, Decomposition of Relational (Lossless and Lossy Decomposition), Conditions to Decomposition, Introduction to Normalization and its Types, Problems on Normalization, Transaction States , ACID Properties, Concurrency Problems in Transactions.

Module 5: Schedules

Types of Schedules, Serializability in DBMS, Problems on Serializability, Recoverability, Cascading Rollback (Cascadeless and Strict Schedules), Equivalence of Schedules, Practice Problems.

Module 6: Overview and Storage of Indexing

Data on External Storage – File Organization and Indexing – Cluster Indexes, Primary and Secondary Indexes – Index data Structures – Hash Based Indexing – Tree base Indexings)

Books:

1. A Silberschatz, H Korth, S Sudarshan, “Database System and Concepts”, fifth Edition McGraw-Hill, Rob, Coronel, “Database Systems”, Seventh Edition, Cengage Learning.

References:

1. DBMS Computer Science Gate Forum, GATE Vidyalaya
2. C. J. Date, An Introduction to Database Systems
3. Ivan Bayross, SQI, PI/SQL The Programing Language of ORACLE

Course Outcome:

After undergoing this course the students will be able to:

1. Understand database concepts and structures and query language
2. Understand the E R model and relational model
3. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.
4. Execute various advanced SQL queries related to Transaction Processing & Locking using the concept of Concurrency control.
5. Understand the principles of storage structure and recovery management.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 353	GRAPH THEORY	3	0	0	70	30	-	100	3

Course Objectives:

1. To learn the fundamental theory about graphs (definitions, theorems and their proofs)
2. To study the basic algorithms of graph theory and their modifications
3. To know applications of graph theory

Module 1: Graphs, Sub graphs, some basic properties, various examples of graphs & their sub graphs, walks, path & circuits, connected graphs, disconnected graphs and components, euler graphs, various operations on graphs, Hamiltonian paths and circuits, the traveling salesman problem.

Module 2: Trees and fundamental circuits, distance diameters, radius and pendant vertices, rooted and binary trees, counting trees, spanning trees, fundamental circuits, finding all spanning trees of a graph and a weighted graph, algorithms of primes, Kruskal and Dijkstra Algorithms.

Module 3: Cuts sets and cut vertices, some properties, all cut sets in a graph, fundamental circuits and cut sets , connectivity and separability, network flows Planer graphs, combinatorial and geometric dual: Kuratowski graphs, detection of planarity, geometric dual, Discussion on criterion of planarity, thickness and crossings.

Module 4: Vector space of a graph and vectors, basis vector, cut set vector, circuit vector, circuit and cut set subspaces, Matrix representation of graph – Basic Concepts; Incidence matrix, Circuit matrix, Path matrix, Cut-set matrix and Adjacency matrix. Coloring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problems.

Suggested Text Book & References:

1. Deo, N, Graph theory with applications to Engineering and Computer Science, PHI.
2. Gary Chartrand and Ping Zhang, Introduction to Graph Theory, TMH.
3. Robin J. Wilson, Introduction to Graph Theory, Pearson Education.
4. Harary, F, Graph Theory, Narosa
5. Bondy and Murthy: Graph theory and application. Addison Wesley.
6. V. Balakrishnan, Schaum's Outline of Graph Theory, TMH

Course Outcomes:

1. Solve problems using basic graph theory
2. Identify induced subgraphs, matchings, covers in graphs
3. Determine whether graphs are Hamiltonian and/or Euclerian
4. Solve problems involving vertex and edge connectivity, planarity
5. Solve problems involving vertex and edge coloring
6. To understand and apply the fundamental concepts in graph theory

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 354	COMPUTER NETWORKS	3	0	0	70	30	-	100	3

Course Objectives:

The objective(s) of this course is to-

1. To understand the concept of data communication and the evolution of computer networks using the layered network architecture.
2. To introduce the student to the major concepts involved in personal area network (PAN), wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).
3. To understand the various error control flow control techniques in the data link layer.
4. To understand the IP address range for subnets and routing protocols used in computer networks.
5. To outline the various functions of transport and application layer protocol.

Detailed contents

Module 1: Introduction:

Data communication Components: Representation of data and its flow Networks , Various Connection Topology, Protocols and Standards, OSI model, TCP/IP Network Model Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

Module 2: Data Link Layer and Medium Access Sub Layer:

Error Detection and Error Correction -Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA.

Module 3: Network Layer:

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.

Module 4: Transport Layer:

Process to Process Communication, User Datagram Protocol (UDP),Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

Module 5: Application Layer:

Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Network Security.

Suggested books :

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. Larry L. Peterson and Bruce S. Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann Publishers., San Francisco, Fifth Edition, 2011

Suggested reference books

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.
4. William Stallings, "Cryptography and Network Security - Principles and Practices", Prentice Hall of India, Seventh Edition, 2017.

Course Outcomes

At the end of the course, the student will be able to:

1. Explain data communication system, components requirement of networks and network topology.
2. Explain the functions of the different layers of the OSI reference model and TCP/IP Network model.
3. Draw the functional block diagram of Personal area network (PAN), wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
4. Explain the data link layer and network layer protocols.
5. Outline the mechanisms involved in the transport layer.
6. Learn different application layer protocols.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 355	ARTIFICIAL INTELLIGENCE	3	0	0	70	30	-	100	3

Course Objective:

The Student should be made to:

1. Study the concepts of Artificial Intelligence.
2. Learn the methods of solving problems using Artificial Intelligence.
3. Introduce the concepts of Expert Systems and machine learning.

Module 1:

Introduction to AI: What is AI ? Examples of AI systems. Approaches to AI. Brief history of AI Intelligent Agent : stimulus-response agents. components of intelligence

Module 2:

Introduction to State Space Search Statement of Search problems: state space graphs. Searching explicit state spaces. Feature based state spaces. Problem types, examples (puzzle problem, n-queen, the road map, traveling salesman, etc.) Uninformed Search: Formulating the state space. Greedy search, breadth-first, depth- first, iterative deepening, bidirectional search Informed Search Strategies I - Using evaluation functions. A general graph searching algorithm. Uniform cost search, A*, admissibility of A* Informed Search Strategies II - Iterative deepening A*, recursive best first search.

Problem Solving using Search -Two agent search Adversarial search- Two agent games. Minimax Two agent games- alpha beta pruning

Module 3:
Constraint satisfaction problems- Definitions, examples, constraint-graph, backtracking, forward checking, constraint propagation (arc-consistency, path-consistency) dynamic ordering, incremental repair (min-conflicts heuristic), CSP and SAT, GSAT

Module 4:
Knowledge Representation and Logic - Propositional Logic Propositional logic, syntax, semantics, semantic rules, terminology - validity, satisfiability, interpretation, entailment, proof systems Propositional Logic inference rules, natural deduction, propositional resolution.

Module 5:
Knowledge Representation and Logic - First Order Logic, First Order Logic – I Motivation, Syntax, Interpretations, semantics of quantifiers First Order Logic- II Entailment in FOL, interpretation Inference in FOL - I First Order resolution. Conversion to clausal form. Inference in FO
Rule based Systems Rule Based Systems - I Forward chaining. Backward chaining. Conflict resolution Rule Based Systems – II

Suggested Text Book & References:

1. Elaine Rich and Kevin Knight, "Artificial Intelligence", McGraw-Hill.
2. Stuart Russell, Peter Norvig, "Artificial Intelligence – A Modern Approach", Pearson Education.
3. E Charniak and D McDermott, "Introduction to Artificial Intelligence", Pearson Education.
4. Dan W. Patterson, "Artificial Intelligence and Expert Systems", Prentice Hall of India.

Course Outcomes:
Upon completion of the course, students will be able to
1. Identify problems that are amenable to solution by AI methods.
2. Identify appropriate AI methods to solve a given problem.
3. Formalise a given problem in the language/framework of different AI methods.
4. Implement basic AI algorithms.
5. Design and carry out an empirical evaluation of different algorithms on a problem formalisation, and state the conclusions that the evaluation supports.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 356	PROGRAM ELECTIVE – I	3	0	0	70	30	-	100	3

CS 356 PROGRAM ELECTIVE -I LIST (Any One of the following)

1. **PROGRAMMING IN PYTHON**
2. **PROGRAMMING IN MATLAB**
3. **PROGRAMMING IN JAVA**
4. **PROGRAMMING IN PHP**

1. PROGRAMMING IN PYTHON

Course Objective:

1. This course is designed to provide basic knowledge of Python.
2. Intended for students with little or no programming experience.
3. It aims to provide students with an understanding of the role of computation and helps in problem solving and programming..

Course Details

Learning Outcomes:

Problem solving and programming capability

Fundamentals of Python- History, Features, Comparison with other languages, Applications, Setting up path, Working with Python, Basic Syntax, Data Types and Variables, Using Numeric Variables, Using String Variables.

Data Handling- Introduction, Data Types-Mutable and Immutable ,Operators, Expressions-Evaluating Expressions and Type Casting, Working with some standard Library Modules-Math Module, Random Module and Statistics Module.

Conditional and Iterative Statement – If Statements of Python, range ( ), Iteration and Looping Statements.

Text Handling – Traversing a String, String Operators, Methods, Slicing.

Lists and Tuple- Creating and Accessing, Working with List and Tuples, Functions and Methods

Dictionaries, Functions and Module-Introduction, Accessing values in dictionaries, Working with dictionaries, Properties of Dictionary, Defining a function, Calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables, Importing module, Math module, Random module, Packages, Composition

Numeric and Date Functions, Exception Handling - Dates and Times, Advanced Data and Time Management, Random Numbers, The Math Library, Character Data, Exception, Exception Handling, Except clause, Try ? finally clause, User Defined Exceptions.

Working with Numpy-Anatomy of Numpy Arrays, Numpy array vs. Python Lists, Numpy Data Types, Creating Numpy Arrays-1 Dimension and 2 Dimension, Accessing Individual Elements using Array Indexing, Array Slices, Joining or Concatenating Numpy Arrays, Arithmetic Operations on 2D Array.

Books Recommended for Reading and Reference:

1. Python-(Mark Lutz)
2. Python Training guide (BPB Publications)
3. Python Programming, Michael Dawson, 3rd Edition, Course technology, 2010
4. Informatics Practices by Sumita Arora

Course Outcome

1. Explain basic principles of Python programming language
2. Implement object oriented concepts.
3. Implement database and GUI applications.

2. PROGRAMMING IN MATLAB

Course Objectives:

Students will be introduced to different inbuilt functions and modules provided by Matlab and Lab View for control applications.

Introduction to MATLAB:

Matlab Interactive Sessions- Menus and the toolbar- Computing with Matlab- Script files and the Editor Debugger- Matlab Help System

Arrays:

Arrays- Multidimensional Arrays- Element by Element Operations-Polynomial Operations Using Arrays- Cell Arrays-Structure Arrays

Functions & Files:

Elementary Mathematical Functions-User Defined Functions-Advanced Function Programming-Working with Data Files

Programming Techniques:

Program Design and Development-Relational Operators and Logical Variables-Logical Operators and Functions-Conditional Statements-Loops- The Switch Structure-Debugging Matlab Programs

Plotting:

XY- plotting functions-Subplots and Overlay plots-Special Plot types- Interactive plotting-Function Discovery-Regression- 3-D plots

Linear Algebraic Equations:

Elementary Solution Methods-Matrix Methods for (LE) - Cramer's Method

Probability and Statistics:

Interpolation -Statistics, Histogram and probability- The Normal Distribution-Random number Generation-Interpolation

Symbolic Processing with Matlab:

Symbolic Expressions and Algebra-Algebraic and Transcendental Equations-Calculus-Symbolic Linear Algebra

Image Processing:

Vector Graphics-Morphological Image Processing-Filtering

Suggested Text Book & References:

1. Hanselman, D. and B. Littlefield, "Mastering MATLAB 7," PEARSON/Prentice Hall.
2. Etter, D.M. and D.C. Kuncicky, "Introduction to MATLAB", E-Source, Prentice Hall.
3. Palm III, W.J., "Introduction to MATLAB for Engineers," B.E.S.T Series, McGraw-Hill.

Course Outcome

At the end of the course, students will be able to

1. Use computational tools to support analysis and synthesis requirements in control.
2. Support documentation in the form of graph, figure, audio-visuals etc.

3. PROGRAMMING IN JAVA

Course Objective:

The objectives of this course are to introduce:

1. Fundamentals of OOP in Java.
2. Concepts of packages and interfaces.
3. Thread and event handling in Java.
4. GUI using applets.
5. Students how Java and a DBMS may work together.

Java Basics: History of Java, Java Versions, A Simple Java Program, JVM Architecture, Data Types, Variables, Arrays, Operators, Control Statements, Type Casting.

Classes And Inheritance: The General Form of a Class, Methods, Constructors, The Finalize( ) Method, Inheritance Concepts, Member Access Rules, Method Overloading, Method Overriding, Abstract Class, Interface, String and String Buffer Classes, The Object Class.

Package And I/O: Defining Package, Finding Packages and CLASSPATH, Access Protection, Importing Packages, Concepts of Streams, Byte and Character Streams, Console Input and Output, Input Stream, Output Stream, File Input Stream, File Output Stream, Reader, Writer, File Reader, File Writer

Exception Handling and Thread: Exception Types, Try, Catch, Throws, Finally, Built-In Exceptions, Concepts Of Thread, Thread Life Cycle, Thread Class and Runnable Interface, Creating Multiple Threads, Synchronization, Thread Priorities.

Applet: Applet Basics, Applet Architecture , An Applet Skeleton, Simple Applet Display Methods, Requesting Repainting, Using the Status Window, APPLET Tag, Passing Parameters to Applets, get Document Base( ), get Code Base( ), Applet Context and show Document( ), Outputting to the Console, Loading an Image, Displaying an Image, Image Observer.

AWT, Swing and Event Handling: Labels, Button, Text Field, Text Area, Check Box, Radio Button, Choice, List, J Button, J Check Box, J Radio Button, J Label, J Text Field, J Text Area, J List, Event Sources, Event Listeners, Action Event Class, Action Listener Interface, Window Event Class, Window Listener Interface, Mouse Event Class, Window Listener Interface, Adapter Classes, Inner Classes, Anonymous Inner Classes.

Introduction to JDBC: JDBC Overview & Architecture, JDBC Driver Types, Executing a simple DML Command.

References:

1. Java-The Complete Reference, Herbert Schildt ,McGraw Hill, 2020.
2. Teach Yourself Java in 21 Days, Rogers Cadenhead, Sams Publishing, 2020.
3. Programming with Java, E Balagurusamy, McGraw-Hill, 2019.

Course Outcomes:

1. Students will be familiar with Encapsulation, Inheritance and Polymorphism.
2. Students will be able to develop GUI-based java programs.
3. Students will be able to connect a DBMS and perform insert, update and delete operations on DBMS tables.

4. PROGRAMMING IN PHP

Course Objectives:

The objectives of this course are to:

1. Familiar with client server architecture and able to develop a web application using PHP.
2. Introduce the skills and project-based experience needed for entry into web application and development careers.

History of internet and World Wide Web, Web server like Apache, Multi-tier Application architecture, client side versus server scripting, accessing web servers, HTML, XHTML, CSS, PHP Essentials, Installation and Configuration files, Variables, constants, Operators, Control Structures, Strings, Array, Functions, Built-in PHP Function Libraries, Forms, Data Validation, File Handling (Including and Requiring Files, Reading and Writing Files, Allowing Users to Download Files), PHP ODBC, Sessions, Cookies, FTP, GET and POST data, HTTP Headers, HTTP Authentication

Books and References

1. Timothy, Elizabeth, Jason, Scouarnec (Author), Jeremy, Michael K. Glass, Beginning PHP6, Apache, MYSQL Web Development, Wiley/Wrox, 2009.
2. PHP 6 and MySQL 5 for Dynamic Web Sites: Visual Quick Pro Guide, Larry Ullman, Pearson Education.
3. Steven Holzner, PHP: The Complete Reference, TMH

Course Outcomes:

1. Students are able to develop PHP programs. .
2. Students will be able to connect a DBMS and perform insert, update and delete operations on the DBMS table.
3. Students will be able to write a server side application to catch form data sent from the client, process it and store it on a database.

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 352P	DATABASE MANAGEMENT SYSTEM LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 352.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 356P	PROGRAM ELECTIVE – I LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 356.

MANDATORY COURSE

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
NC 351	ORGANIZATIONAL BEHAVIORS	3	0	0	-	*50	-		0

Each individual has behavior patterns that are shaped by the context of his or her past. Most often adapting the behavior to the changing context of the reality a person lives in becomes difficult which may lead to the reduction in personal effectiveness and natural self-expression. The main focus of this course is to equip the students with useful approaches to help in the deeper understanding of self and help individuals empower themselves to be the source of their own growth and development. The course will help students to learn effective communication skills, group and team building skills and will help them learn the goal setting process and thus become more effective in achieving their goals. The broader objective of this course is to make the students aware about the different facets of self and to help them learn skills to strengthen their inner capacities. So that they are able to understand themselves, think and act effectively, to be able to communicate in an effective manner and to learn to lead and to form an effective team.

The specific objectives, however, are as follows:

1. To help the students to understand their real self by recognizing different aspects of their self-concept that will lead to increased self-confidence.
2. To train the students for communicating effectively in both formally as well as informal settings.
3. To help the students to understand the importance of non-verbal aspects of effective communication.
4. To help the students to understand emotion and emotional intelligence, managing ones' own emotional reservoirs, effective dealing with emotions at work.
5. To facilitate the students in understanding the formation and function of group and team and to help them to learn the skills of a successful leader.
6. To help the students in understanding and practicing the goal setting process by recognizing the importance of each step involved in goal setting. The activities involved are designed to facilitate their career goal decision making.

The activities to achieve the above objectives can be suggested as follows:

1. Motivational lectures
2. Group discussion/activities
3. Case study
4. Games / Stimulation exercises
5. Role-playing
6. Mindfulness training

DETAILED SYLLABUS (SIXTH SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 361	COMPILER DESIGN	3	0	0	70	30	-	100	3

Course objectives

The objective(s) of this course is to:

1. Learn the fundamentals of the Design of Compilers by applying mathematics and engineering principles
2. Construct lexical analyser and the parsing methods that are typically used in compilers
3. Know about the principle ideas in syntax directed definitions and translations to generate intermediate code for the typical programming languages.
4. Understand about the run time environment, code generation and code optimization.

Detailed contents

Module 1:

Introduction: Language Processors, The structure of Compiler, Evolution of programming Languages, Application of compiler technology Compilers and Interpreters - The structure of a Compiler. Phases of compilation and overview.

Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, from regular expressions to finite automata.

Module 2:

Syntax Analysis (Parser): Definition - role of parsers, Top down parsing: elimination of Left recursion and left factoring, FIRST- FOLLOW, Recursive descent parsing, Bottom-up parsing: Handle pruning, Shift reduce parsing, LR parsing – LR (0) items - SLR parsing, Canonical LR parsing.

Module 3:

Semantic Analysis: Syntax directed translation schemes- Synthesized and Inherited attributes, Dependency graph, L-attributed and S-attributed definition, Implementation of syntax directed translators,

Symbol Table: Its structure, symbol attributes and management.

Module 4:

Run-time environment: Storage Organization, Activation Record, Parameter passing, value return, Storage Allocation Strategies, scope.

Intermediate Code Generation: Intermediate Languages - prefix - postfix - Quadruple - triple - indirect triples, Syntax tree- Evaluation of expression - three-address code, Translation of assignment statement - Boolean expressions, Type Checking.

Module 5:

Code Generation: Issue in the design of a code generator, Analysis: control-flow, data-flow dependence, Basic Blocks and Flow Graph, Code Generation algorithm, code generation from DAG.

Code optimization: Machine dependent and independent optimization, principal sources of optimization, Optimization of Basic Blocks, loop optimization, peep-hole optimization etc.

Course Outcomes

At the end of the course, the student will be able to:

1. Develop the lexical analyser for a given grammar specification
2. Design top-down and bottom-up parsers for a given parser specification
3. Translate into various intermediate codes and develop syntax directed translation schemes.
4. Apply various rules for designing and generating code for compiler design.
5. Analyse different optimization methods on intermediate code to generate efficient compilers.

Text Books:-

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman "Compilers: Principles, Techniques and Tools", Pearson Education, Inc., Second Edition, 2013.
2. Alfred V. Aho and Jeffrey D. Ullman, "Principles of Compiler Design", Narosa Publishing House Pvt. Ltd., Twenty Fifth Reprint 2002.
3. S. Godfrey Winster, S. Aruna Devi, R. Sujatha, "Compiler Design", Yes Dee Publishing Pvt. Ltd, 2016.

Reference Books:-

1. Jean Paul Tremblay and Paul G. Sorenson., "Theory and Practice of Compiler Writing", BS Publication, Reprint 2008.
2. K. Muneeswaran, "Compiler Design", Oxford Higher Education, Fourth edition 2015.
3. David Galles, "Modern Compiler Design", Pearson Education, Reprint 2012.
4. Raghavan V., "Principles of Compiler Design", Tata McGraw Hill Education Pvt. Ltd., 2010.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 362	SOFTWARE ENGINEERING	3	0	0	70	30	-	100	3

Course Objectives:

1. Knowledge of basic SW engineering methods and their appropriate application.
2. A general understanding of software process models such as the waterfall and evolutionary models.
3. Understanding of software requirements and the SRS documents.
4. Understanding of approaches to verification and validation including static analysis, and reviews.
5. Understanding of software testing approaches such as unit testing and integration testing.
6. Describe software measurement and software risks.
7. Understanding on quality control and how to ensure good quality software.

Module-1: Introduction:

What is Software Engineering-Importance of software - the software evolution - software characteristics – software components-software crisis and its solutions.

Software Development Life-cycle:

Requirements analysis, software design, coding, testing, maintenance, etc.

Module-2: Software Requirements Specification:

Waterfall model, prototyping, interactive enhancement, spiral model- Role of Management in software development- Role of metrics and measurement-Problem analysis, requirement specification, validation, metrics, monitoring and control.

Module-3: System Design:
Problem partitioning, abstraction, top-down and bottom-up design, Structured Approach. Functional versus object-oriented approach, design specification and verification metrics, monitoring and control, cohesion, coupling.

Module-4: Coding :
Top-down and bottom-up, structured programming, information hiding, programming style, and internal documentation. Verification, Metrics, monitoring and control.

Module-5: Testing:
Black box and white box testing. Unit testing, integration testing, system testing techniques to generate test plans. Mathematical methods of software verification. Alpha and Beta testing. Verification and Validation.

Module-6: Software Project Management:
Cost estimation, Project scheduling, Staffing, Software configuration management, Quality Assurance, Project Monitoring, Risk management, etc.

Suggested Text Book & References:

1. R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill
2. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
3. Pankaj Jalote, Software Engineering, Wiley
4. Carlo Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI.
5. Ian Sommerville, Software Engineering, Addison Wesley.
6. Pfleeger, Software Engineering, Pearson Education.

Course Outcomes:
1. Analyze the requirements
2. Categorize requirements and design SRS
3. Apply software engineering principles and techniques.
4. Design and evaluate large-scale software systems.
5. Demonstrate ethical standards and legal responsibilities.
6. Identify suitable process model for a given software requirement

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 363	PROGRAM ELECTIVE – II	3	0	0	70	30	-	100	3

LIST OF PROGRAM ELECTIVE – II

1. **DATA COMMUNICATION**
2. **FORMAL METHODS FOR SYSTEM VERIFICATION**
3. **HIERARCHICAL MEMORY ALGORITHM**
4. **INTRODUCTION TO COMPLEXITY THEORY**
5. **LOGIC FOR COMPUTER SCIENCE**
6. **PARALLEL ALGORITHM**

1. DATA COMMUNICATION

Course Objective-

1. Students are expected to learn basics of Communication Technologies and data communication which will help them to build fundamentals for learning Computer Networks in higher semester.
2. The course is designed to let students demonstrate an understanding of the fundamentals of data communication, types of transmission mediums and interfacing standards along with current edge of the data compression techniques.

UNIT - I: Introduction to data communication: Components , data representation ,data flow and basic model ,data representation, Serial & Parallel transmission, Modes of data transmission, Encoding: Unipolar, Polar, Bipolar line & block codes, Data compression Frequency dependent codes, Run length encoding, Relative encoding, LZ Compression Image and multimedia compression. Review of analog & digital transmission methods, Nyquist Theorem.

UNIT-2: Multiplexing: FDM, TDM, WDM, Synchronous & Statistical TDM, North American digital multiplexing hierarchy, European TDM, Spread spectrum: Frequency Hopping & Direct Sequence spread spectrum. Terminal handling & polling. Switched Communication Networks: Circuit, Message, Packet & Hybrid Switching, Softswitch Architecture with their comparative study, X.25, ISDN.

UNIT-3: Physical Layer: Introduction, Interface, Standards, EIA-232-D, RJ-45, RJ-11, BNC connector & EIA-449 digital Interface: Connection, specifications & configuration, X.21 Modem: Types, features, signal constellation, block schematic, limited distance, dial up, baseband, line driver, Group Band and Null modems etc., ITU-T V-series modem standards Connecting Devices: Active and Passive Hubs, Repeaters, Bridges, Two & Three layer switches & Gateway. Study of various types of topology and their comparative study and introduction to queuing theory.

UNIT-4: Transmission Media: Transmission line characteristics, distortions, Crosstalk, Guided Media: Twisted Pair, Baseband & Broadband Coaxial. Optical Fibre : Physics and velocity of propagation of light , Advantages & Disadvantages, Block diagram, Nodes And classification ,Comparison, losses, light source and detectors, Construction, Unguided media: Electromagnetic polarization, Rays and waves front, electromagnetic spectrum and radiation, spherical wave front and inverse square law, wave attenuation and absorption, optical properties of Radio waves, Terrestrial Propagation of electromagnetic waves, skip distance, free - space path loss, Radio waves , Microwave, Infrared & Satellite Communication system. Telephone Network: Components, LATAs, signalling and Services, Digital Subscriber Line: ADSL, HDSL, SDSL, VDSL, Cable TV network for data transfer.

UNIT-5: Transmission Errors : Content Error , flow integrity error , methods of error control, Error detection, Error correction, Bit error rate, Error detection methods: Parity checking, Checksum Error Detection, Cyclic Redundancy Check, Hamming code, Interleaved codes, Block Parity, Convolution code, Hardware Implementation, Checksum.

Text Books/References

1. I. A. Dhotre, V.S. Bagad, Data Communication, Technical Publications Pune, 2008.
2. A.S. Tanenbaum, Computer Networks, Pearson Education.
3. W. Stallings, Data and Computer Communication, Pearson Education.
4. Forouzan, "Data Communication and Networking", TMH.
5. G. Shanmugaratnam, "Essential of TCP/ IP", Firewall Media

Course Outcomes

After completion of this course, student will be able to

1. Understand the importance of data communication systems and fundamentals.
2. Distinguish and relate various physical Medias, interfacing standards and adapters.
3. Explain various flow control techniques.
4. Analyze various modulation technique in analog and digital system
5. Understand Physical layer of LAN, MAN and WAN
6. Analyze short range and long range wireless technologies

2. FORMAL METHODS FOR SYSTEM VERIFICATION

Course objectives

1. Learn about formal modeling and specification languages
2. Write and understand formal requirement specifications
3. Learn about main approaches in formal system verification
4. Know which formal methods to use and when
5. Use automated and interactive tools to validate models and code

Introduction to formal methods and hardware verification. Review of logics: Propositional Calculus and Predicate Calculus. Axioms and rules of Floyd-Hoare Logic. APPLICATION of Floyd-Hoare logic to verify hardware circuits. Describing hardware directly in higher ORDER logic. Combinational and sequential behaviour of circuits. Specification of hardware systems. Introduction to Binary Decision Diagram (BDD) and modelling hardware with BDDs. Algorithms for BDD operations. Concept of OBDDs and ROBDDs and operation on ROBDDs. Introduction to Temporal Logic. Linear and Branching time temporal logic. Expressing properties in CTL and CTL*. CTL model CHECKING algorithm. State space explosion problem: Symbolic data structure and symbolic model checking algorithms. Concept of on-the-fly model checking and automata-theoretic model checking. Study of verification tools: SMV and PVS.

Books and References

1. M. Huth and M. Ryan, Logic in COMPUTER SCIENCE: Modelling and Reasoning about Systems, 2nd Ed, Cambridge University PRESS, 2004.
2. T. F. Melham, Higher Order Logic and Hardware Verification, Cambridge University Press, 1993.
3. E. M. Clarke, O. Grumberg and D. Peled, Model Checking, MIT Press, 1999.
4. K. L. McMillan, Symbolic Model Checking, Kluwer Academic Publisher, 1993.
5. Z. Manna and A. Pnueli, The Temporal Logic of Reactive and Concurrent System Specification, Springer-Verlag, 1992.

Course Outcomes:

1. Design correct programs to solve problems
2. Choose efficient memory algorithms and apply them to solve problems.
3. Analyze the efficiency of programs based on time complexity
4. Able to apply efficient algorithm in various areas like computational geometry

3. HIERARCHICAL MEMORY ALGORITHMS

Course Objectives

1. To introduce the fundamental concept memory and its level
2. It emphasize the importance of memory in developing and implementing efficient algorithms
3. To understand the external memory data structure and use of cache algorithms.

Hierarchical memory levels; performance characteristics; Parallel disk model. Fundamental I/O operations. Design and analysis of efficient external memory algorithms for some representative problems. Sorting, permutation, searching. Depth first search, breadth first search, Minimum spanning forest, connected components, single source shortest path, transitive closure. hashing, string matching. External Memory Data Structures. Cache efficient algorithms. APPLICATIONS in various areas, for example, Computational geometry.

Text Books and References

1. J. S. Vitter. Algorithms and Data Structures for External Memory, Now Publishers, 2008.
2. N. Zeh. I/O-efficient graph algorithms, Lecture notes from EEF Summer School on Massive Data Sets, 2002.
3. L. Arge. The Buffer Tree: A New Technique for Optimal I/O Algorithms, Proceedings of the fourth International Workshop on Algorithms and Data Structures (WADS), 1995.
4. M. H. Nodine and J. S. Vitter. Greed Sort: Optimal Deterministic Sorting on Parallel Disks, Journal of the ACM (JACM), 42 (4), 1995.
5. R. D. Barve, E. F. Grove and J. S. Vitter. Simple Randomized Mergesort on Parallel Disks, Proceedings of the eighth ACM symposium on Parallel algorithms and architectures (SPAA), 1996.
6. A. Beckmann, U. Meyer, P. Sanders and J. Singler. Energy-efficient Sorting using Solid State Disks, Proceedings of the first International Green Computing Conference (IGCC), 2010.

Course Outcomes:

1. Design correct programs to solve problems
2. Choose efficient memory algorithms and apply them to solve problems.
3. Analyze the efficiency of programs based on time complexity
4. Able to apply efficient algorithm in various areas like computational geometry

4. INTRODUCTION TO COMPLEXITY THEORY

Course Objectives:

1. To learn the Big-O's of some standard algorithms.
2. To learn the theory of computational complexity and standard complexity classes.
3. To learn the classification of computational problems and know how difficult they are to solve.
4. To use the Turing machine to solve a problem.

Turing Machines, Turing Machine Construction, The Halting Problem, Un decidability, Recursive functions, Church-Turing thesis ,Machines and languages, Reducibility between languages, Complexity classes: The class P, The robustness of the class P, Polynomial-time reducibility, The class NP, NP-complete languages, Computing and verifying a function, Optimization problems, Reducibility and optimization problems, Beyond NP, The class co-NP, The Boolean hierarchy, The polynomial hierarchy, Exponential-time complexity classes, Space-complexity classes, Space-complexity classes, relations between time and space, determinism and space, Non determinism, complement and space, Logarithmic space, Polynomial space, Probabilistic algorithms and complexity classes, Some probabilistic algorithms, Probabilistic Turing machines, Probabilistic complexity classes.

Text Books/References

1. Sanjeev Arora, Boaz Barak, Computational Complexity: A Modern Approach Hardcover ,Cambridge University Press; 1 edition,2009.
2. Oded Goldreich , Computational Complexity: A Conceptual Perspective, Cambridge University Press; 1 edition,2008.

Course Outcome:

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

1. Classify decision problems into appropriate complexity classes.
2. Reduce one problem to another, and construct reductions for simple examples.
3. Classify optimization problems into appropriate approximation complexity classes.
4. To construct a Turing machine for a particular problem.

5. LOGIC FOR COMPUTER SCIENCE

Course Objective:

To learn the fundamental role of logic in computer science.

1. To learn the main notions of mathematical logic- logical notations (syntax) and how to assign meaning to them (semantics).
2. To learn some uses for mathematical logic in the field of computer science.
3. To learn the construction of logical arguments (proof theory).

Communication and Concurrency: Processes as transition systems, operations on these processes (composition, hiding etc.). Bisimulation and observational equivalences. Calculus of mobile systems: pi-calculus. Some theories related to pi calculus. Logics to reason about transition systems, LTL, CTL* and modal mu calculus.

Reasoning about Knowledge: Knowledge as modality, axioms of knowledge. Common knowledge, distributed agents exchanging messages, agreeing to disagree. Logical omniscience.

Finite Model Theory: Expressiveness of FO and its extensions on finite structures. Games for lower bounds. Connections with complexity classes, role of order on the domain.

Feasible Proofs: Propositional proof systems for tautologies. Simulation and lower bounds on length of proofs for specific systems (e.g. PHP requires super polynomial length using resolution). Theories of weak arithmetic, probably total functions and relations to complexity theory.

Full Abstraction problem for PCF: PCF as an extension of lambda calculus. Operational and denotational semantics and the full abstraction problem. Solutions to the full abstraction problem. Games semantics.

Text Books/References

1. R. Milner, Communicating and Mobile Systems: the pi calculus, Cambridge University press, 1999.
2. R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, Reasoning about Knowledge, MIT press, 1995.
3. H. D. Ebbinghaus, J. Flum, Finite Model Theory, Springer, 1995.
4. N. Immerman, Descriptive Complexity, Springer, 1999.
5. J. Krajicek, Bounded Arithmetic, Propositional Logic and Complexity Theory, Cambridge university press, 1995.

Course Outcomes

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

1. Know Propositional and First-order Logic.
2. Know how logic may be used in specifying properties. Classify optimization problems into appropriate approximation complexity classes.
3. Know how to do a proof by induction, and some important meta-theorems about deduction systems.

6. PARALLEL ALGORITHM

Course Objectives:

1. To make the students familiar with Parallel Computation and techniques for parallelization
2. To enable students understand how to reduce the number of processors and calculating cost of communication
3. To give knowledge about parallel search, elementary parallel algorithm, graph algorithm, P-complete classes.
4. To enable students to understand the concept of Mutual exclusion and Clock Synchronization, Distributed Graph algorithms.
5. To make the students understand basics of Cover MPI programming

Introduction: Paradigms of parallel computing: Synchronous - vector/array, SIMD, Systolic; Asynchronous - MIMD, reduction paradigm. Hardware taxonomy: Flynn's classifications, Handler's classifications. Software taxonomy: Kung's taxonomy, SPMD.

Abstract parallel computational models: Combinational circuits, Sorting network, PRAM models, Interconnection RAMs. Parallelism approaches - data parallelism, control parallelism

Performance Metrics: Laws governing performance measurements. Matrices - speedups, efficiency, utilization, communication overheads, single/multiple program performances, bench marks.

Parallel Processors: Taxonomy and topology - shared memory multiprocessors, distributed memory networks. Processor organization - Static and dynamic interconnections. Embeddings and simulations.

Parallel Programming: Shared memory programming, distributed memory programming, object oriented programming, data parallel programming, functional and dataflow programming.

Scheduling and Parallelization: Scheduling parallel programs. Loop scheduling. Parallelization of sequential programs. Parallel programming support environments.

Text Books and References

1. Henri Casanova, Arnaud Legrand, Yves Robert, Parallel Algorithms, CRC Press, 2008.
2. M.J. Quinn, "Designing Efficient Algorithms for Parallel Computers" by Mc Graw Hill.
3. S.G. Akl, Design and Analysis of Parallel Algorithms, Prentice Hall
4. S.G. Akl, Parallel Sorting Algorithm, Academic Press

Course Outcomes:

1. The students will be able to understand basics of PRAM Model of Parallel Computation, techniques for parallelization like pointer jumping and Divide and Conquer
2. The students would be able to perform preorder traversal and understand parallel computing platforms and find the cost of communication.
4. The students will be able define elementary parallel algorithms and Dictionary operations
5. The students will be able to measure complexity,
6. The students would be able to demonstrate Parallel Monte Carlo and write simple programs using MPI programming

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 364	PROGRAM ELECTIVE – III	3	0	0	70	30	-	100	3

LIST OF PROGRAM ELECTIVE – III

1. **PROGRAMMING IN JAVASCRIPT**
2. **PROGRAMMING IN R**
3. **PROGRAMMING IN GO**
4. **PROGRAMMING IN ANDROID**
5. **PROGRAMMING IN XML**

1. PROGRAMMING IN JAVASCRIPT

Course Objective:

The objectives of this course are to:

1. Familiar with client server architecture and able to develop a web application using JavaScript technologies.
2. Introduce the skills and project-based experience needed for entry into web application and development careers.

Introduction, Simple program, obtaining user input with prompt dialogs, memory concepts, arithmetic, decision making, assignment operators, control structures – IF, IF…ELSE, WHILE, FOR repetition statement, SWITCH multiple-selection statement, logical operators, Program modules in JavaScript, function definitions, scope rules, global functions, recursion, arrays, references and reference parameters, passing arrays to functions, sorting arrays, searching arrays, multidimensional arrays, math object, string object, date object, Boolean and number object, document object, window object, using cookies.

Text Books and References

1. Mastering JavaScript and Jscript, James Jaworski, Sybex.
2. Teach Yourself JavaScript in 21 Days, Jonathan A. Watt, Sams,2002

Course Outcomes:

1. Students are able to develop JavaScript programs.
2. Students are able to develop a dynamic webpage by the use of javascript and DHTML.
3. Students will be able to write a client side application to process form data before sending it to a server.

2. PROGRAMMING IN R

Course Objectives:

The objectives of this course are to:

1. Understand the Basics in R programming in terms of Constructs, Control Statements, String Functions.
2. Understand the use of R for Big Data analytics.
3. Apply R programming for Text Processing.
4. Appreciate and Apply the R programming in Statistics.

Introduction: Introducing to R – R Data Structures – Help functions in R – Vectors – Scalars – Declarations – recycling – Common Vector operations – Using all and any – Vectorized operations – NA and NULL values – Filtering – Vectorised if-then else – Vector Equality – Vector Element names.

Matrices, Arrays And Lists: Creating matrices – Matrix operations – Applying Functions to Matrix Rows and Columns – Adding and deleting rows and columns – Vector/Matrix Distinction – Avoiding Dimension Reduction – Higher Dimensional arrays – lists – Creating lists – General list operations – Accessing list components and values – applying functions to lists – recursive lists.

Data Frames: Creating Data Frames – Matrix-like operations in frames – Merging Data Frames – Applying functions to Data frames – Factors and Tables – factors and levels – Common functions used with factors – Working with tables - Other factors and table related functions - Control statements – Arithmetic and Boolean operators and values – Default values for arguments - Returning Boolean values – functions are objects – Environment and Scope issues – Writing Upstairs - Recursion – Replacement functions – Tools for composing function code – Math and Simulations in R.

OOP: S3 Classes – S4 Classes – Managing your objects – Input/Output – accessing keyboard and monitor – reading and writing files – accessing the internet – String Manipulation – Graphics – Creating Graphs – Customizing Graphs – Saving graphs to files – Creating three-dimensional plots.

Interfacing: Interfacing R to other languages – Parallel R – Basic Statistics – Linear Model – Generalized Linear models – Non-linear models – Time Series and Auto-correlation – Clustering.

Text Books and Reference Books:

1. Norman Matloff, The Art of R Programming: A Tour of Statistical Software Design, No Starch Press, 2011.
2. Jared P. Lander, R for Everyone: Advanced Analytics and Graphics, Addison-Wesley Data & Analytics Series, 2013.
3. Mark Gardener, Beginning R – The Statistical Programming Language, Wiley, 2013.
4. Robert Knell, Introductory R: A Beginner's Guide to Data Visualisation, Statistical Analysis and Programming in R, Amazon Digital South Asia Services Inc, 2013.

Course Outcomes:
After successful completion of the course students should be able to:-
1. Understand the basics in R programming in terms of constructs, control statements, string functions.
2. Understand the use of R for Big Data analytics.
3. Learn to apply R programming for Text processing.
4. Able to appreciate and apply the R programming from a statistical perspective.

PROGRAMMING IN GO
Course Objectives:
1. This course will teach you the basic principles of Go programming language which is used extensively with Linux, MacOS, and Windows platforms.
2. This course will help developers to create real-world applications which has an extensive set of libraries.

Course Contents:
Introduction, Variables, Data Types, Arithmetic, Constants, Strings, Booleans, Printf, Logical Operators, For Loops, Relational Operators, If, Switch, Arrays, Slice, Maps, Functions, Closures, Recursion, Defer, Recover, Pointers, Structs, Interfaces, File I/O., Casting, Create Web Server, Go Routines, Web Scraping.

Text Books and Reference Books:
1. The Go Programming Language (Addison-Wesley Professional Computing Series), Alan A. A. Donovan and Brian W. Kernighan, Pearson Education, 2015
2. An Introduction to Programming in Go, Caleb Doxsey, Create Space Independent Publishing Platform, 2012
3. The Go Programming Language Phrasebook, David Chisnall, Pearson Education, 2012.
4. Learning Go Programming, Vladimir Vivien, Packt Publishing, 2016
5. The Way to Go- A Thorough Introduction to the Go Programming Language, Ivo Balbaert, I Universe Publication, 2012.

Course Outcomes:
After completion of this course, student will be able to
1. Know the basic principles of Go programming language.
2. Be able to create your own stand-alone command-line apps.
3. Will be able to create Scripts Network and Web servers.
4. Understand Go Routines and Channels.
5. Enhance your aptitude through innovative and independent learning.

3. PROGRAMMING IN ANDROID
Course Objective:
1. This course facilitates classroom and laboratory learning, letting students develop competence and confidence in android programming.
2. To understand the entire Android Apps Development Cycle.
3. It would also enable the students to independently create Android Applications

UNIT 1-Basic of Android Programming:
Introduction to Android OS, Setting up the Android Application Development Environment, Creating, Testing and Debugging Applications, Android Stack, Android applications structure, Activity life cycle, Understanding implicit and explicit intents.

UNIT II-User Interface in Android: Adaptive and responsive user interfaces, User Input Controls, Menus, Screen Navigation, Recycler View, Drawables, Themes and Styles, Fragments Fragment Life Cycle, Introduction to Material Design, Testing the user interface.

UNIT III-Background tasks: Async Task, AsyncTask Loader, Connecting App to Internet, Broadcast receivers, Services, Notifications, Alarm managers.

UNIT IV-Sensor, Location and Maps: Sensor Basic, Motion and Position Sensors, Location services, Google maps API, Google Places API.

UNIT V-Working with data in Android: Shared Preferences, App Setting, SQLite primer, Store data using SQLite database, Content Providers, Content Resolver, Loader.

UNIT VI- Performance Improvement of App: Performance Parameters, Profiling Tools, Rendering and Layout, Garbage Collection and Memory Leaks, Best Practices.

UNIT VII-Publishing Your App: Preparing for publishing, Signing and preparing the graphics, Publishing to the Android Market.

Textbooks and References

1. Android: A Programming Guide by J.F. Di Marzio.
2. Hello, Android: Introducing Google's Mobile Development Platform by Ed Burnett.
3. Programming android by Zigurd Mednieks
4. Android User Interface Design: Turning Ideas and Sketches into Beautifully Designed Apps by Ian G. Clifton.
5. Android Developer Fundamental Course by Google.
6. Advance Android Developer Course by Google.

Course Outcomes:

After completion of this course, student will be able to

1. Demonstrate the Understanding of fundamentals of Android Programming.
2. Build their ability to develop software with reasonable complexity on a mobile platform.
3. Discover the life cycles of Activities, Applications, intents and fragments.
4. Design the Android apps by using Java Concepts.

5. PROGRAMMING IN XML

Course Objective:

The objectives of this course are to

1. Describe Well-Formed XML.
2. Define XML
3. Identify Document Type Definitions
4. List the different types of XML Schemas.
5. Apply XML to well-constructed documents for Web browser-based technology in business information systems.

XML Introduction and Overview, XML Syntax, XML Namespaces and Infoset, Document Type Definitions (DTDs), XML Schemas, Data Modeling, XPATH, XSL, XSLT, XLink, and X Pointer, Document Object Model (DOM), Simple Application Programmer Interface (API) for XML (SAX), XML and Databases, Simple Object Access Protocol (SOAP)

Books and References

1. Williamson, Xml: The Complete Reference, TMH,2008.
2. Steven Holzne, Sams Teach Yourself XML in 21 Days, 3rd Edition, SAMS.
3. Devan Shepherd, Sams Teach Yourself XML in 21 Days, 2001, SAMS
4. Bill Evjen, Kent Sharkey, Thiru Thangarathinam, Michael Kay, Alessandro Vernet, Sam Ferguson, Professional XML, Wrox/Peer Information Inc.; 2nd, 2000.

Course Outcomes:

Upon successful completion of the course, students should be able to:

1. Describe how XML can be applied to well-constructed documents for Web browser-based technology in business information systems
2. Identify concepts related to connecting resources with links, CSS, DTD, and internationalization
3. Design and apply XML to create a markup language for data and document centric application
4. Develop well-formed web browser-based documents utilizing XML
5. Develop code to demonstrate understanding of knowledge related to XML

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 365	OPEN ELECTIVE – I	3	0	0	70	30	-	100	3

LIST OF OPEN ELECTIVE – I

1. **INFORMATION THEORY & CODING**
2. **DATA ANALYSIS AND PROCESS INTERPRETATION**
3. **EMBEDDED SYSTEMS**
4. **FUZZY LOGIC**
5. **ARTIFICIAL NEURAL NETWORKS**

1. INFORMATION THEORY & CODING

Course Objectives

1. To define and apply the basic concepts of information theory (entropy, channel capacity etc.)
2. To learn the principles and applications of information theory in communication systems.
3. To study various data compression methods and describe the most common such methods.
4. To understand the theoretical framework upon which error-control codes are built.

Communication processes. Channel matrix. Probability relation in a channel. The measure of information. Entropy function – Properties of entropy function. Channel capacity. Special types of channels. Binary symmetric channel. Encoding. Block code. Binary code. Binary Huffman code. Shannon–Fano Coding procedure. Noiseless coding theorem. Shannon's first theorem. Error – correcting codes. Examples of codes. Hadamard matrices and codes. Binary Golay code. Matrix description of linear codes. Equivalence of linear codes. The Hamming codes. The standard array. Syndrome decoding.

Text Books/References:

1. Monica Borda, Fundamentals in Information Theory and Coding, 2011, Springer.
2. Wells Richa Wells Richard B, Applied Coding and Information Theory for Engineers, Pearson Education, 2009.
3. Ranjan Bose, Information Theory, Coding and Cryptography, McGraw Hill, 2008.
4. Steven Roman, Coding and Information Theory, Springer.
5. Hsu & Mitra, Analog and Digital Communication, Schaum's Series, 2008.

Course Outcome

Up on successful completion of this course a student will be able to:

1. quantify the notion of information in a mathematically sound way
2. explain what is the significance of this quantitative measure of information in the communications systems
3. calculate entropy, joint entropy, relative entropy, conditional entropy, and channel capacity of a system
4. differentiate between lossy and lossless compression techniques
5. decide an efficient data compression scheme for a given information source
6. explain the impact of feedback and/or many senders or receivers on the communication systems

2. DATA ANALYSIS AND PROCESS INTERPRETATION

Course Objectives:-

The main purpose of data analysis is to find meaning in data so that the derived knowledge can be used to make informed decisions. The implementation of processes through which data is reviewed for the purpose of arriving at an informed conclusion. The interpretation of data assigns a meaning to the information analyzed and determines its signification and implications.

The role of statistics, Graphical and numerical methods for describing and summarising data. Probability, population distributions. Sampling variability and sampling distributions. Estimation using a single sample. Hypothesis testing a single sample. Comparison of two populations or treatments. Simple linear regression and correlation. Case studies.

Text Books/References:

1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 3rd Edition, Elsevier, New Delhi.
2. Papoulis, Probability, Random Variables, and Stochastic Processes, 4th Edition, McGraw Hill.
3. William Feller, An Introduction to Probability Theory and Its Applications, 3rd Edition, Wiley India.

Course Outcome:

Should understand the data analysis involved in working to uncover patterns and trends in datasets and also know that data interpretation involves explaining those patterns and trends. Must know that Scientists interpret data based on their background knowledge and experience; thus, different scientists can interpret the same data in different ways.

3. EMBEDDED SYSTEMS

Course objectives:

1. To introduce the Building Blocks of Embedded Systems and microcontrollers.
2. To Educate in Various Embedded Development Strategies.
3. To Introduce Bus Communication in processors, Input/output interfacing.
4. To introduce Basics of Real time operating systems.

UNIT I: EMBEDDED COMPUTING

Challenges of Embedded Systems – Embedded system design process. Embedded processors – 8051 Microcontroller, ARM processor – Architecture, Instruction sets and programming.

UNIT II: MEMORY AND INPUT / OUTPUT MANAGEMENT

Programming Input and Output – Memory system mechanisms – Memory and I/O devices and interfacing – Interrupts handling.

UNIT III: PROCESSES AND OPERATING SYSTEMS

Multiple tasks and processes – Context switching – Scheduling policies – Interprocess communication mechanisms – Performance issues.

UNIT IV: EMBEDDED SOFTWARE

Programming embedded systems in assembly and C – Meeting real time constraints – Multi-state systems and function sequences. Embedded software development tools – Emulators and debuggers.

UNIT V: EMBEDDED SYSTEM DEVELOPMENT

Design issues and techniques – Case studies – Complete design of example embedded systems.

Books and References

1. Lyla, Embedded Systems, Pearson, 2013
2. Frank Vahid, Tony Givargis, Embedded System Design - John Wiley.
3. David E. Simon, An Embedded Software Primer - Pearson Education.
4. Raj Kamal, Embedded Systems - TMH.

Course outcomes:

1. On the successful completion of the course, students will be able to
2. Understand the concept of embedded system, microcontroller, and different components of microcontroller.
3. Acquire a basic knowledge about fundamentals of Embedded Systems and microcontrollers.

4. FUZZY LOGIC

Course Objectives

The Student should be made to:

1. Get the exposure to Fuzzy Logic.
2. Understand the importance of tolerance of imprecision and uncertainty for design of robust & low cost intelligent machines.

Course Contents:

Introduction to fuzzy sets. Fuzzy sets and basic definitions. Extensions. Fuzzy measures and measures of fuzziness. The extension principle and applications. Fuzzy relations and fuzzy graphs. Fuzzy Logic, Fuzzy Sets and System Modelling, Fuzzy Relations, Fuzzy Equations, Membership Function in Fuzzy Logic, Fuzzy Rules, Fuzzy analysis. Fuzzy set theory. Decision making in fuzzy environments. Fuzzy set models in operations research. Empirical examination in fuzzy set theory. Applications with Matlab Fuzzy Toolbox.

Text Books and References

1. Timothy J. Rose, Fuzzy Logic with Engineering Applications, Wiley India.
2. Samir Hung T. Nguyen, Elbert A. Walker, A First Course in Fuzzy Logic, Third Edition, CRC Press.
3. Samir Roy, Udit Chakraborty, Introduction to Soft Computing: Neuro-Fuzzy and Genetic Algorithms, 1st Edition, Pearson Education, India.
4. S. Rajasekaran, G. A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic And Genetic Algorithm: Synthesis And Applications, PHI, 2011.

Course Outcomes:

Upon completion of the course, students will be able to

1. Identify and describe Fuzzy Logic techniques in building intelligent machines
2. Apply Fuzzy Logic models to handle uncertainty and solve engineering problems
3. Recognize the feasibility of applying a Fuzzy model for a particular problem

5. ARTIFICIAL NEURAL NETWORKS

Course Objectives

Students will try to learn:

1. To introduce some of the fundamental techniques and principles of neural computation.
2. To investigate some common models and their applications.

Course Content

Overview of Biological Neurons: Structure of biological neurons relevant to ANNs.

Fundamental Concepts of Artificial Neural Networks: Models of ANNs; Feed forward & feedback networks; learning rules; Hebbian learning rule, perception learning rule, delta learning rule, Widrow-Hoff learning rule, correction learning rule, Winner –lake all learning rule, etc.

Single layer Perceptron Classifier: Classification model, Features & Decision regions; training & classification using discrete perceptron, algorithm, single layer continuous perceptron networks for linearly separable classifications.

Multi-layer Feed forward Networks: Linearly non-separable pattern classification, Delta learning rule for multi-perceptron layer, Generalized delta learning rule, Error back-propagation training, learning factors, Examples.

Single layer feedback Networks: Basic Concepts, Hopfield networks, Training & Examples.

Associative memories: Linear Association, Basic Concepts of recurrent Auto associative memory: retrieval algorithm, storage algorithm; By directional associative memory, Architecture, Association encoding & decoding, Stability.

Self organizing networks: Unsupervised learning of clusters, winner-take-all learning, recall mode, Initialisation of weights, separability limitations.

Textbooks and References:

1. Mohamad H. Hassoun, ”Fundamentals of Artificial Neural Networks”
2. Laurene Fausett, "Fundamentals of Neural Networks" , Pearson Education, 2004..
3. Simon Haykin, "Neural Networks- A comprehensive foundation", Pearson Education, 2003.
4. S.N. Sivanandam, S.Sumathi, S. N. Deepa "Introduction to Neural Networks using MATLAB 6.0", TATA Mc Graw Hill, 2006.
5. S. Rajasekharan and G. A. Vijayalakshmi pai, "Neural Networks, Fuzzy logic, Genetic algorithms: synthesis and applications", PHI Publication, 2004.
6. Timothy J. Ross, "Fuzzy Logic With Engineering Applications", Tata McGraw- Hill Inc. 2000

Course Outcomes:-

On completion of this course, a student should be able to:

1. Understand the learning and generalisation issue in neural computation.
2. Understand the basic ideas behind most common learning algorithms for multilayer perceptrons, radial-basis function networks, and Kohonen self-organising maps.
3. Implement common learning algorithms.
4. Apply neural networks to classification and recognition problems.

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 361P	COMPILER DESIGN LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 361.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 362P	SOFTWARE ENGG. LAB.	0	0	4	-	30	70	100	2

Laboratory practical based on CS 362.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 366P	MINI PROJECT	0	0	8	-	60	140	200	3

Details of the mini project are to be decided by the Department.

DETAILED SYLLABUS (SEVENTH SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 471	PROGRAM ELECTIVE –IV	3	0	0	70	30	-	100	3

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 472	PROGRAM ELECTIVE –V	3	0	0	70	30	-	100	3

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 473	PROGRAM ELECTIVE –VI	3	0	0	70	30	-	100	3

LIST OF PROGRAM ELECTIVE-IV, PROGRAM ELECTIVE-V, PROGRAM ELECTIVE-VI

1. **COMPUTATIONAL NUMBER THEORY**
2. **DATA MINING**
3. **MACHINE LEARNING**
4. **NETWORK PROTOCOLS**
5. **LINUX INTERNAL**
6. **RANDOMIZED ALGORITHMS**
7. **INFORMATION RETRIEVAL**
8. **HUMAN COMPUTER INTERACTION**
9. **COMPUTER GRAPHICS**
10. **CLOUD COMPUTING**
11. **ADVANCED C**
12. **PROBABILITY THEORY AND RANDOM PROCESSES**
13. **SOFTWARE PROJECT MANAGEMENT**
14. **SOFTWARE RELIABILITY**
15. **DISTRIBUTED ALGORITHMS**
16. **DISTRIBUTED DATABASE SYSTEM**

1. COMPUTATIONAL NUMBER THEORY

Course Objectives

1. To understand congruences, residue classes and least residues.
2. To determine multiplicative inverses in modulo n and use them to solve linear congruences.
3. To solve a quadratic congruence.
4. To find out whether an integer is a prime using Miller-Rabin test.
5. To factorize an integer using the Pollard rho method.
6. To implement the Rabin Cryptosystem.
7. To state and prove Fermat's Little Theorem and its generalisation using Euler's function & use them to implement the RSA.

Course Content:

Unit-I: Arithmetic of Integers-Additive Inverse- Multiplicative Inverse-gcd- Modular Arithmetic- Modular Exponentiation-Linear Congruences-Chinese Remainder Theorem-Polynomial Congruences- Quadratic Residues-Modular Square Roots.

Unit-II: Representation of Finite Fields- Prime and Extension Fields, Representation of Extension Fields-Polynomial Basis-Irreducible Polynomials.

Unit-III: Algorithms for Polynomials- Root Finding and Factorization, Polynomials over Finite Fields.

Unit-IV: Primality Testing Algorithms-Fermat Test, Miller-Rabin Test, AKS Test.

Unit-V: Integer Factoring Algorithms-Trial Division, Pollard rho Method, Quadratic Sieve Method.

Unit-VI: Application in Cryptography-RSA-Rabin Cryptosystem

Course Outcomes

Students will be able to :

1. Understand the basics of modular arithmetic.
2. Solve systems of Diophantine equations using the Chinese Remainder Theorem & the Euclidean algorithm.
3. Solve a quadratic congruence.
4. Test primality using Miller-Rabin test.
5. State and prove Fermat's Little Theorem & its generalisation using Euler's function & use them to implement the RSA.
6. Implement Rabin Cryptosystem.
7. To know the problems of RSA.

Textbooks & References:

1. Das, Computational Number Theory, CRC Press.
2. David Burton, Elementary Number Theory, McGraw Hill
3. V. Shoup, A computational introduction to number theory and algebra, Cambridge University Press.
4. G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers, Oxford University Press.

2. DATA MINING

Course Objective:

1. Be familiar with mathematical foundations of data mining tools..
2. Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
3. Master data mining techniques in various applications like social, scientific and environmental context.
4. Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

Introduction: Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective.

Data Mining Techniques: A Statistical Perspective on Data Mining, Similarity Measures, Decision Trees, Neural Networks, Genetic Algorithms.

Classification: Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree-Based Algorithms, Neural Network-Based Algorithms, Rule-Based Algorithms, Combining Techniques.

Clustering: Similarity and Distance Measures, Hierarchical Algorithms, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes.

Association Rules : Basic Algorithms, Parallel and Distributed Algorithms, Incremental Rules, Advanced Association Rule Techniques, Measuring the Quality of Rules.

Advanced Techniques: Web Mining, Spatial Mining, Temporal Mining.

Books and References:

1. J. Han & M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2nd Edn, 2006.
2. M. J. A. Berry and G. Linoff, Mastering Data Mining: The Art and Science of Customer Relationship Management, Wiley, 2000.
3. P. Adriaans & D. Zantinge, Data Mining, Addison Wesley, 1996.
4. R. Mattison, Data Warehousing: Strategies, Tools and Techniques, McGraw Hill, 1996.
5. P. Ponniah, Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals, Wiley, 2001.

Course Outcomes:

Upon successful completion of this course, you should be able to

1. Understand the functionality of the various data mining and data warehousing component.
2. Explain the analyzing techniques of various data.
3. Describe different methodologies used in data mining.
4. Compare different approaches of data warehousing and data mining with various technologies.
5. Evaluate the performance of different data-mining algorithms.
6. Propose data-mining solutions for different applications.

3. MACHINE LEARNING

Course Objectives:

1. Makes best use of data to maximise business opportunities.
2. Acquire advanced Data Analysis skills.
3. Create AI/ML solutions for various business problems.

Data Preprocessing:

Importing the dataset, Missing Data, Categorical Data, Splitting the dataset into the Training set and Test set, Feature Scaling.

Regression: Simple Linear Regression, Multiple Linear Regression, Polynomial Regression, SVR, Decision Tree Regression, Random Forest Regression, Evaluating Regression Models Performance.

Classification: Logistic Regression, K-Nearest Neighbors (K-NN), Support Vector Machine (SVM), Kernel SVM, naïve Bayes, Decision Tree Classification, Random Forest Classification, Evaluating Classification Models Performance.

Clustering: K-Means Clustering, Hierarchical Clustering.

Association Rule Learning and Reinforcement Learning: Apriori, Eclat, Upper Confidence Bound (UCB), Thompson Sampling, Natural Language Processing, Deep Learning(Artificial Neural Networks, Convolutional Neural Networks).

Dimensionality Reduction and Model Selection & Boosting

Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), KernelPCA, k-Fold Cross Validation, GridSearch, XG Boost.

Textbooks and References:

1. Stephen Marsland, Machine Learning: An Algorithmic Perspective
2. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 2nd Edition, 2010.
3. Mitchell, Machine Learning. TMH, 2013.
4. Jason Bell, Machine Learning for Big Data: Hands-on for Developers and Technical Professionals, Wiley India, 2014.
5. Christopher Bishop, Pattern Recognition and Machine Learning, 2010, Springer.

Course Outcomes:

1. Gain knowledge about basic concepts of Machine Learning
2. Identify machine learning techniques suitable for a given problem
3. Solve the problems using various machine learning techniques
4. Apply Dimensionality reduction techniques.
5. Design application using machine learning techniques.

4. NETWORK PROTOCOLS

Course Objectives:

1. To learn the different network architectures and protocols.
2. To learn the various TCP/IP protocols.
3. To learn the various network security technologies and protocols.
4. To understand VOIP protocols.
5. To understand WAN and LAN protocols.

Overview of IPv4, TCP, IPv6, ICMP, ARP, DHCP; Routing Protocols: OSPF, RIP, BGP, Ad hoc network routing (AODV, DSR); IP Security: NAT, IPSEC, Socks, SSL; Quality of Service related protocols: Intserv, diffserv, Queuing techniques (WFQ, RED, etc.); Multi-Protocol Label Switching (MPLS) and GMPLS; Virtual Private Network (VPN) Protocols: L2TP, PPTP; Overview of Application Layer Protocols: DNS, LDAP, SMTP, POP3, IMAP4, SNMP; Voice over IP Protocols (VOIP) and videoconferencing: SIP, H323. Server Load BALANCING Techniques.

Text Books and References

1. Charles. M.Kozierok, TCP/IP GUIDE, Shroff Publishers, Mumbai, 2005.
2. Uyless Black, MPLS and Label Switching Networks, Pearson Education (LPE), 2002
3. Forouzan, TCP/IP Protocol Suite, 3/E, TMH, 2008.
4. Karanjit Siyan, Tim Parker, TCP/IP Unleashed, 2002, Sams.
5. Request for Comments (RFC) from www.ietf.org.

Course Outcomes

On completion of the course, the students should be able to:

1. Understand the different network architectures and protocols.
2. Design different TCP/IP protocols.
3. Understand various network security technologies and protocols.
4. Understand VOIP protocols.
5. Understand the WAN and LAN protocols.

5. LINUX INTERNAL

Course Objectives

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

1. To understand the fundamentals of Linux.
2. To know the Kernel data structures.
3. To understand processes in Linux.
4. To learn different Linux system calls.

UNIT- I: GENERAL OVERVIEW OF THE SYSTEM

History − System structure − User perspective − Operating system services − Assumptions about hardware − Introduction to the kernel − Architecture of the LINUX operating system − Introduction to system concepts − Kernel data structures − System administration − Summary and preview.

UNIT- II: BUFFER CACHE

Buffer headers − Structure of the buffer pool − Advantages and disadvantages of the buffer cache − Internal representation of files − Inodes − Structure of a regular file − Directories − Conversion of a path name to an inode − Superblock − Other file types.

UNIT- III: SYSTEM CALLS FOR FILE SYSTEM

Open − Read − Write − File and record locking − Adjusting the position of file I/O − LSEEK − Close − File creation − Creation of special files − Pipes − Dup − Mounting and unmounting file systems.

UNIT- IV: THE STRUCTURE OF PROCESSES

Process states and transitions − Layout of system memory − The context of a process − Saving the context of a process − Process control − Process creation − Signals − Process termination − Awaiting process termination − Invoking other programs − The shell − System boot and the INIT process.

UNIT- V: PROCESS SCHEDULING AND MEMORY MANAGEMENT POLICIES

Process scheduling − Memory management policies − Swapping − A hybrid system with swapping and demand paging − The I/O subsystem − Driver interfaces − Disk Drivers − Terminal drivers.

Text Books/References

1. Bach Maurice, The Design of the Unix Operating System Paperback, PHI.
2. Robert Love, Linux Kernel Development, Pearson,2010
3. Richard Petersen, Linux: The Complete Reference, TMH, Sixth Edition, 2007

Course Outcomes

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

1. Understand the fundamentals of Linux.
2. Know the INIT and Kernel data structures.
3. Analyze different Linux system calls.
4. To create processes by fork ( ) system calls.

6. RANDOMISED ALGORITHMS

Course Objectives

Upon completion of this course, students will be able to do the following:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Review of discrete probability; Notion of randomized algorithms, motivating examples; Markov, Chebyshev inequalities, Chernoff bounds; Probabilistic method; Hashing, fingerprinting; Random walks and Markov chains. Program checkers; Polynomial identities; Randomized complexity classes, probabilistically checkable proofs; some number theoretic problems; approximate counting.

Books and References

1. Rajeev Motwani, Prabhakar Raghavan, Randomized Algorithms Paperback, Cambridge University Press,2008.
2. Michael Mitzenmacher, Eli Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis,2005.
3. Russ Bubley, Randomized Algorithms: Approximation, Generation, and Counting, Springer.
4. Ketan Mulmuley, Computational Geometry: An Introduction Through Randomized Algorithms, Prentice-Hall.

Course Outcomes

Students who complete the course will have demonstrated the ability to do the following:

1. Argue the correctness of algorithms using inductive proofs and invariants.
2. Analyze worst-case running times of algorithms using asymptotic analysis.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it like Markov etc
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it
5. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.
6. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.
7. Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.

7. INFORMATION RETRIEVAL

Course Objective:

The Student should be made to:

1. Learn the information retrieval models.
2. Be familiar with Web Search Engine.
3. Be exposed to Link Analysis.
4. Understand Hadoop and Map Reduce.
5. Learn document text mining techniques.

Course Contents:

1. Introduction to Information Retrieval: Motivation-Information Retrieval vs Data Retrieval-Flashback
2. Models of Information Retrieval: Boolean Model-Vector Space Model-Probabilistic Model-Alternative Models
3. Retrieval Evaluation: Recall and Precision-Alternative Measures-Reference Collections and Evaluation of IR systems
4. Query Languages for IR: Keywords-Boolean Queries-Context Queries-Natural Language Queries-Structural Queries
5. Advanced Query Operations: Relevance Feedback-Query Expansion-Automatic Local Analysis-Automatic Global Analysis

6. Text Indexing, Preprocessing and File Organization: Stopwards, stemming, thesauri-File (Text) organization (invert, stuff)-Text statistics (properties)-Text compression
7. Text Searching: Knuth-Morris-Pratt-Boyer-Moore family-Suffix automaton-Phrases and Proximity
8. Document Clustering & Multimedia Information Retrieval: Similarity Queries-Feature-based Indexing and Searching-Spatial Access Methods-Searching in Multidimensional Spaces
9. Parallel and Distributed IR: Architectures MIMD and SIMD-Collection Partitioning-Source
10. Selection-Query Processing--2-Peer Architectures and Systems
11. Meta-Ranking: Integrated vs Isolated Methods-Interleaving-Voting
12. Web Search: History of Web-Indexing-Spidering/Crawling-Link Analysis (HITS, PageRank)
13. User Interfaces and Visualization.

Textbooks and References

1. Baeza-Yates, Berthier- Neto, Modern Information Retrieval, Pearson, 2009.
2. Gerald J. Kowalski, Information Retrieval Systems: Theory and Implementation, Springer, 2007.
3. Frakes, Information retrieval: data structures & algorithms, 2009, PHI.
4. Gerald J. Kowalski, Mark T. Maybur, Information Storage and Retrieval Systems: Theory and Implementation, Springer, 2002.

Course Outcomes:

Upon completion of the course, students will be able to

1. Apply information retrieval models.
2. Design Web Search Engine.
3. Use Link Analysis.
4. Use Hadoop and Map Reduce.
5. Apply document text mining techniques

8. HUMAN COMPUTER INTERACTION

Course Objective:

The Student should be made to:

1. Learn the foundations of Human Computer Interaction.
2. Be familiar with the design technologies for individuals and persons with disabilities.
3. Be aware of mobile HCI.
4. Learn the guidelines for user interface.

HCI foundation: HISTORY, human abilities, state of the art in computing technology, interaction styles and paradigms; Design process: interaction design basics, HCI in software process, design rules and guidelines, implementation support (UI software), universal design; Interaction styles: direct manipulation, WIMP, web interface, natural language interaction; Evaluation techniques; Models in HCI: formal models, linguistic models, cognitive models (KLM/GOMS), cognitive architectures, hybrid models; Task analysis; Dialogue design; Advanced topics (overview) pervasive computing, CSCW, virtual reality, tangible USER interface, multimedia.

Textbooks and References

1. A. Dix, J. Finlay, G. D. Abowd and R. Beale, Human Computer Interaction, 3rd Edition, Pearson Education, 2005
2. J. Preece, Y. Rogers, H. Sharp, D. Baniyon, S. Holland and T. Carey, Human Computer Interaction, Addison-Wesley, 1994.

3. C. Stephanidis (ed.), User Interface for All: Concepts, Methods and Tools. Lawrence Erlbaum Associates, 2001.
4. J. M. Caroll (ed.), HCI Models, Theories and Frameworks: TOWARDS a Multidisciplinary Science (Interactive Technologies), Morgan Kauffman, 2003.
5. W. O Galitz, The Essential Guide to User Interface Design, John Wiley & Sons, Inc, 2002 (Indian Edition).
6. B. Shneiderman, Designing the User Interface, Addison Wesley, 2000 (Indian Reprint).

Course Outcomes:

Upon completion of the course, students will be able to

1. Design effective dialog for HCI.
2. Design effective HCI for individuals and persons with disabilities.
3. Assess the importance of user feedback.
4. Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Websites.
5. Develop a meaningful user interface.

9. COMPUTER GRAPHICS

Course Objectives:

Students will

1. Learn the basics of various inputs and output computer graphics hardware devices.
2. Learn the fundamental concepts in 2D and 3D computer graphics.
3. Learn 3D modeling, geometric transformations, 3D viewing and rendering.
4. Design and implement practical graphic solutions to challenging problems in different application domains.

Detailed Syllabus

Graphics display devices, Input and output devices, Raster Graphics: Line and Circle drawing algorithms, Windowing, Clipping: Cohen and Sutherland line clipping, Cyrus-beck clipping method, 2D and 3D Geometrical Transformations, Viewing Transformations: parallel and perspective projection, Curves and Surfaces: Cubic splines, Bezier curves, B-splines,, Fractal curves and surfaces, Hidden line/surface removal methods, Illumination model, Polygon Shading: Gouraud and Phong.

Suggested Text Books & References:

1. Zhigang Xiang, Schaum's Outline of Computer Graphics, 2nd Edition, TMH.
2. Donald Hearn and M Pauline Baker, “Computer Graphics C Version”, Pearson Education.
3. Rogers, “Procedural Elements of Computer Graphics”, McGraw Hill
2. Amrendra N Sinha and Arun D Udai,” Computer Graphics”, TMH
3. Steven Harrington, “Computer Graphics: A Programming Approach”, TMH

Course Outcome:

After learning the course the students should know:

1. The various computer graphics hardware and display technologies.
2. 2D and 3D viewing technologies
3. Various 2D and 3D objects transformation techniques.
4. The underlying algorithms, mathematical concepts, supporting computer graphics.

10. CLOUD COMPUTING

Course Objectives:

The key objective of this course is to make students:

1. Understand the fundamentals and essentials of Cloud Computing.
2. Understand the key security and compliance challenges of cloud computing
3. Understand the key technical and organizational challenges
4. Understand the different characteristics of public, private and hybrid cloud deployment models.
5. Understand a sound foundation of Cloud Computing so that they are able to start using and adopting Cloud Computing services and tools in their real life scenarios.
6. Explore some important cloud computing driven commercial systems and applications.
7. Expose to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

Unit-1: Cloud Architecture and Model:

Technologies for Network-Based System – System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture. Cloud Models:- Characteristics – Cloud Services – Cloud models (IaaS, PaaS, SaaS) – Public vs Private Cloud –Cloud Solutions - Cloud ecosystem – Service management – Computing on demand.

Unit-2: Virtualization:

Basics of Virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Virtual Clusters and Resource management – Virtualization for Data-center Automation.

Unit-3: Cloud Infrastructure:

Architectural Design of Compute and Storage Clouds – Layered Cloud Architecture Development – Design Challenges - Inter Cloud Resource Management – Resource Provisioning and Platform Deployment – Global Exchange of Cloud Resources.

Unit-4: Programming Model:

Parallel and Distributed Programming Paradigms – Map Reduce , Twister and Iterative Map Reduce – Hadoop Library from Apache – Mapping APPLICATIONS - Programming Support - Google App Engine, Amazon AWS - Cloud Software Environments - Eucalyptus, Open Nebula, Open Stack, Aneka, Cloud Sim

Unit-5: Security in the Cloud:

Security Overview – Cloud Security Challenges and Risks – Software-as-a- Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security.

Suggested Text Book & References:

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
2. John W. Rittinghouse and James F. Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
3. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", TMH, 2009.
4. Rajkumar Buyya, Christian Vecchiola, S. Tamarai Selvi, 'Mastering Cloud Computing", TMGH, 2013.
5. Gautam Shroff, Enterprise Cloud Computing, Cambridge University Press, 2011
6. Michael Miller, Cloud Computing, Que Publishing, 2008.

7. Ronald L. Krutz, Russell Dean Vines, "Cloud Security – A comprehensive Guide to Secure Cloud Computing", Wiley – India, 2010.
8. Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, "Grid and Cloud Computing – A Business Perspective on Technology and Applications", Springer.
9. George Reese, "Cloud Application Architectures: Building APPLICATIONS and Infrastructure in the Cloud" O'Reilly.
10. Kumar Saurabh, "Cloud Computing – insights into New-Era Infrastructure", Wiley India, 2011

Course Outcomes:

After successful completion of this course, students will be able to

1. Explain and analyze various cloud computing models and apply them to solve problems on the clouds.
2. Apply the various fundamental concepts in data centres to understand the tradeoffs in power, efficiency and cost.
3. Identify resource management fundamentals, i.e., resource abstraction, sharing and sandboxing.
4. Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing

11. ADVANCED C

Course Objective:

1. The course is oriented to those students who want to exploit the low-level programming features of C.
2. Students will understand the underlying principle behind low-level data representations, computing, and programming.
3. They will be able to program C with assembly language for the x86 architecture, including register operations, control structures, bitwise operations, and subprograms.
4. Finally, students gain enough hands-on experience with developing a small operating system.

Unit-1:

C Program Compilation, Execution Process, Tokens of C Program, C Instructions, Constants, Variables, Identifiers and Keywords, Primitive Data Types, Structures - The Definition, Structures – Declaration & Type, Accessing Elements of Structure, Range of Signed/Unsigned Data-types, Efficient way of Printing Pointer, Compiler Memory Allocation for Data Types, Compiler Memory Allocation for Structures, Data-type Alignments, Compiler Memory Allocation for Unions, Practical Usage of Unions, Practical Usage of Bit fields, Bit fields Overflow, Printing every byte of an Integer, Enumeration, Types of Statements, Practical example of Types of Usage, type defining a Function Pointer, Bit-Fields in Structure, Practical examples of Bit field Usage, Structure Padding & Pitfalls, Programming Model & Memory Sizes, Why Sizeof int and Long is 4 or 8?, Use of long in 32-bit Architecture, Array – Representation, Array – Memory Allocation, Array – Declaration & Initialization, Two Dimensional Arrays

Unit-2:

Accessing a Variable Through Pointer, Pointer – Memory Allocation, Pointer – Declaration & Initialization, Pointer – Dereferencing Pointers & Arrays, Character Arrays using Pointers, Array of Character Pointers, Memory Diagram – Array of Char Pointers Arrays as Pointers, Constant Pointers, Pointer Arithmetic, String Handling Functions, String Conversion Functions, Efficient usage of sscanf()/sprintf()

Unit-3:
Binary & Octal Systems, Decimal & Hexadecimal Systems, Signed Representations in Memory, Binary Shifts – Right & Left, Sign Bits and Bit- Shift Operations, Right Shift – Logical Vs Arithmetic Shift, Bit-Shift Overflow, ASCII Representations, Endianness – Little Vs Big, Endian-ness –Portability Issues, mixing 80x86 code with c inline code, separate assembly code subroutines, TASM, MASM

Unit-4:
Bitwise Operations, Logical Operators – Short Circuit, Bitwise Vs Logical Operations, sizeof() operator, Pitfalls/Issues with sizeof() usage, Pointer Increment & Scaling, Operator Precedence, Operator Associativity, True meaning of Associativity, Examples of Precedence & Associativity, Ternary Operator Associativity Rule, Data-type Conversion Rules, float to Int to Float Conversions, Variadic functions & default promotion rules, Pointer Format Specifiers, Signed Vs Unsigned – Pitfalls, Evaluation of i = ++i + ++I, Evaluation of i = ++i + ++i + ++i

Unit-5:
Storage Classes, Storage Class Specifiers, Scope of a Variable, Register, Auto, Static, Extern, Why Register Class and Practical Examples, Automatic Variables and Stack, Static Variables and Functions, True meaning of Extern, How to Use extern across Multiple Files with Examples, Best Practices for Extern Usage, Local/Block/Global Scope, Nesting of Scope, Lifetime of a Variable, Linkage of a Variable, What is Const?, Practical Examples of Const Qualifier, Practical Examples of Volatile Qualifier, Register Vs Volatile Performance, Practical Examples of Const Volatile, Pointer Aliasing, Restrict Qualifier,: Dynamic Memory Allocations, malloc, calloc, realloc, malloc Vs calloc, Heap Memory, Stack Memory – Pitfalls, Dangling Pointers, DMA – Errors, Best Practices for malloc() & free(), DMA – Unspecified Behaviour

Unit-6:
Functions & Pointers, Invoking Functions, Passing Arguments to Functions, Call by Value & Reference, Is C call by Value?, Is C call by Reference?, Array as Function Argument, Rules for Array Argument Passing, Multidimensional Array Argument Passing, Structure as Function Argument, Static Vs Dynamic Runtime Environment, Function Call and Runtime Stack, Rules for Evaluation of Function Arguments.

Unit-7:
Memory Organization, Code Segment, Data Segment, Heap Segment, Stack Segment, free space, register space, Stack Frames, Calling Sequence, View of Runtime Stack with Example, Access to Local Variable in Stack, Local Temporaries, Function Pointers, Declaration and Usage of Function Pointers, Function Pointers as Function Parameters, Practical Example of Function Pointers, Pointer to an Integer Array C ,complex pointer declarations, int **p, int (*p)(), int (*p)[], int *p(), int *(*p[])(), int *(**p)[]

Unit-8:
Preprocessor, #include statements, Multiple Inclusion of a Header File?, Preprocessor – #define statements, Conditional Compilation, Preprocessor – Nested Macros, Preprocessor – Multiline Macros, Preprocessor –Stringizer, Preprocessor – Token Concatenation, Preprocessor – Useful Directives, Conditional Directives for Debugging, Where Macros are Heavily Used, Practical Examples of Macros, Macros Pitfalls, Macros Vs Enums, Inline Functions, Macros Vs Inline, Inline Recursive Functions, Command Line Argument, Environment Variables in C Programs, Recursion Example, Recursion Vs Iteration, Code/Space/Time Complexity.

Unit-9:
Standard I/O Library, Files & Streams, Streams Buffers, IO Buffers – Line Vs Full Vs No-Buffers, Setting & Flushing Buffers, File Access, File Access Modes, Sequential Vs Random Access, Concept of File Offsets, File Operation Errors, End-of-File Condition?, Return Values and Error Values, Character Based File I/O, Line Based File I/O, Formatted File I/O, Block File I/O,Dangerous – gets() Vs fgets(),File Random Access Methods.

Textbooks & References

1. C: The Complete Reference b*y Herbert Schildt*
2. Expert C Programming: Deep Secrets by *Peter Van Der Linden*
3. Low-Level Programming: C, Assembly, and Program Execution by *Igor Zhirkov*

Course Outcomes:
Upon successful completion of the course, students should be able to:

1. Exploit many low-level programming features of C.
2. Understand the low-level data representations, computing, and programming.
3. Program C with assembly language.
4. Develop a small operating system.

12. PROBABILITY THEORY AND RANDOM PROCESSES

Course Objectives:

To expose the students to the basics of probability theory and random processes essential for their subsequent study of analog and digital communication.

Course Contents:
Sets and set operations, Probability space, Conditional probability and Bayes theorem, Combinatorial probability and sampling models; Discrete random variables, probability mass function, probability distribution function, example random variables and distributions; Continuous random variables, probability density function, probability distribution function, example distributions; Joint distributions, functions of one and two random variables, moments of random variables; Conditional distribution, densities and moments, Characteristic functions, Markov, Chebyshev and Chernoff bounds, Detection and Estimation. Random sequences and modes of convergence (everywhere, almost everywhere, probability, distribution and mean square), Limit theorems, Strong and weak laws of large numbers, central limit theorem. Random processes, Stationary processes, Mean and covariance functions, Ergodicity, Linear filtering of random processes, Power spectral density.

Text Books/References

1. T Veerarajan, Probability, Statistics and Random Processes, McGraw Hill Education, 2008.
2. Palaniammal S., Probability and Random Processes, PHI, 2011.
3. Himanshu Chaurasiya, Dr. K.M. Soni, Probability, Random Variables and Stochastic Methods, S.K. Kataria & Sons; 2014 edition.
4. Aitsahlia, Chung, Elementary Probability Theory: With Stochastic Processes and An Introduction To Mathematical Finance, Springer 2004.

Course Outcome
Students will be able to

1. Understand the axiomatic formulation of modern Probability Theory and think of random variables as an intrinsic need for the analysis of random phenomena.

2. Characterize probability models and function of random variables based on single & multiples random variables.
3. Evaluate and apply moments & characteristic functions and understand the concept of inequalities and probabilistic limits.
4. Understand the concept of random processes and determine covariance and spectral density of stationary random processes.
5. Demonstrate the specific applications to Poisson and Gaussian processes and representation of low pass and band pass noise models.

13. SOFTWARE PROJECT MANAGEMENT

Course Outcomes:

1. This course is aimed at introducing the primary important concepts of project management related to managing software development projects.
2. They will also get familiar with the different activities involved in Software Project Management.
3. They will also come to know how to successfully plan and implement a software project management activity, and to complete a specific project in time with the available budget.

UNIT-I: Introduction and Software Project Planning: Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan, Software project estimation, Estimation methods, Estimation models, Decision process.

UNIT-II: Project Organization and Scheduling: Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques, Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts.

UNIT-III: Project Monitoring and Control: Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Desk Checks, Walkthroughs, Code Reviews, Pair Programming.

UNIT-IV: Project Management and Project Management Tools : Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, Change Control, Change Requests Management, Version Control, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, Cost Benefit Analysis, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

Books and References

1. Shashikant A. Kelkar, Software Project Management: A Concise Study 2nd Edition, PHI.
2. Bob Hughes, Mike Cotterell, Software Project Management,2010, TMH
3. Pankaj Jalote, Software Project Management in Practice, Pearson Education.

Course Outcomes

1. Identify the different project contexts and suggest an appropriate management strategy.
2. Practice the role of professional ethics in successful software development.
3. Identify and describe the key phases of project management.
4. Determine an appropriate project management approach through an evaluation of the business context and scope of the project.

14. SOFTWARE RELIABILITY

Course Objectives

1. To learn various engineering techniques to develop and maintain reliable software systems.
2. To measure the reliability of a software system.
3. To understand fault prevention, fault removal, fault tolerance and failure forecasting in software systems.
4. To learn different software reliability models and design reliability models for software systems.

UNIT – 1: INTRODUCTION TO RELIABILITY ENGINEERING

Reliability-Repairable and Non Repairable systems-Maintainability and Availability-Designing for higher reliability-Redundancy-MTBF - MTTF - MTTR.

UNIT – 2: SOFTWARE RELIABILITY

Software reliability-Software reliability Vs Hardware reliability-Failures and Faults - Classification of Failures-Counting-System Configuration-Components and Operational Models-Concurrent Systems-Sequential Systems-Stand by Redundant systems.

UNIT – 3: SOFTWARE RELIABILITY APPROACHES

Fault Avoidance - Passive Fault detection - Active Fault Detection - Fault Tolerance - Fault Recovery - Fault Treatment.

UNIT – 4: SOFTWARE RELIABILITY MODELING

Introduction to Software Reliability Modeling – Parameter Determination and Estimation - Model Selection – Markovian Models – Finite and Infinite failure category Models - Comparison of Models – Calendar Time Modeling.

Books and References

1. John Musa, “Software Reliability Engineering”, McGraw-Hill
2. Fenton, and Pfleeger, “Software Metrics: A Rigorous and Practical Approach”, International Thomson Computer Press
3. Jeff Tian, Software Quality Engineering (SQE), Wiley
4. Stephen H. Kan, Metrics and Models in Software Quality Engineering, Addison-Wesley
5. Glenford J. Myers, “Software Reliability “, Wiley Interscience Publication, 1976.

Course Outcomes

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

1. Develop reliable software systems.
2. Understand the fault handling and failure forecasting techniques in software systems.
3. Understand different software reliability models.
4. Design reliability models for software systems.

15. DISTRIBUTED ALGORITHMS

Course Objective:

This course is designed to make it easy for users to access remote resources, and to share them with other users in a controlled manner.

Models of distributed computing, Synchrony, COMMUNICATION and failure concerns, Synchronous message-passing distributed systems, Algorithms in systems with no failures - Leader Election and Breadth-First SEARCH algorithms, The atomic commit PROBLEM, consensus problems - the Byzantine Generals Problem, Asynchronous message-passing distributed systems, Failure detectors, Logical time and vector clocks, Routing algorithms.

Text Books and References

1. Nancy A. Lynch, Distributed Algorithms, Morgan Kaufmann, 1996.
2. Nancy A. Lynch, Distributed Algorithms, Elsevier India; 1st Edition, 2005.
3. Professor Ajay D. Kshemkalyani, Professor Mukesh Singhal, Distributed Computing South Asian Edition: Principles, Algorithms, and Systems, Cambridge University Press, 2010.
4. Wan Fokkink, Distributed Algorithms: An Intuitive Approach, MIT Press, 2013.

Course Outcome :

1. Understand that no central processor is available to handle the calculations.
2. Understand that when a large network must forward all measurement data to a single central processor, there is a communication bottleneck and higher energy drain at and near the central processor.
3. Distributed algorithms for cooperative localization generally fall into one of two schemes.

16. DISTRIBUTED DATABASE SYSTEM

Course Objective:

The aim of the course is to

1. Enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficiencies of the centralized database systems.
2. Introduce basic principles and implementation techniques of distributed database systems.
3. Expose active and emerging research issues in distributed database systems and application development, and 4) apply theory to practice by building and delivering a distributed database query engine, subject to remote Web service calls.

UNIT-I:

Transaction and schedules, Concurrent Execution of transaction, Conflict and View Serializability, Testing for Serializability, Concepts in Recoverable and Cascadeless schedules.

UNIT- II:

Lock based protocols, timestamp based protocols, Multiple Granularity and Multiversion Techniques, Enforcing serializability by Locks, Locking system with multiple lock modes, architecture for Locking scheduler.

UNIT III:
Distributed Transactions Management, Data Distribution, Fragmentation and Replication Techniques, Distributed Commit, Distributed Locking schemes, Long duration transactions, Moss Concurrency protocol.

UNIT- IV:
Issues of Recovery and atomicity in Distributed Databases, Traditional recovery techniques, Log based recovery, Recovery with Concurrent Transactions, Recovery in Message passing systems, Checkpoints, Algorithms for recovery line, Concepts in Orphan and Inconsistent Messages.

UNIT- V:
Distributed Query Processing, Multiway Joins, Semi joins, Cost based query optimization for distributed database, Updating replicated data, protocols for Distributed Deadlock Detection, Eager and Lazy Replication Techniques References.

Textbooks and References:

1. Ozsu M. Tamer, Principles of Distributed Database Systems, Pearson Education, 2006
2. Ramakrishna and Gehrke,' Database Management System, McGraw Hill
3. Garcia-Molina, Ullman,Widom,' Database System Implementation' Pearson Education
4. Ceri and Pelagatti,'Distributed Database', TMH
5. Chhanda Ray, Distributed Database Systems, Pearson 2009.
6. Silberschatz, orth and Sudershan, Database System Concept', Mc Graw Hill

Course Outcomes:
After the completion of the course, the students are expected to
1. Get familiar with the currently available models, technologies for and approaches to building distributed database systems and services.
2. Have developed practical skills in the use of these models and approaches to be able to select and apply the appropriate methods for a particular case
3. Be aware of the current research directions in the field and their possible outcomes
4. Be able to carry out research on a relevant topic, identify primary references, analyze them, and come up with meaningful conclusions.
5. Be able to apply learned skills to solving practical database related tasks.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 474	OPEN ELECTIVE – II	3	0	0	70	30	-	100	3

LIST OF OPEN ELECTIVE II (ANY ONE)

1. **CYBER LAW AND ETHICS**
2. **INTERNET-OF-THINGS**
3. **MULTIMEDIA SYSTEMS**
4. **RESEARCH METHODOLOGY**
5. **GENETICS ALGORITHM**

1. CYBER LAW & ETHICS

Course Objectives:

1. The objective of this course is to enable learners to understand, explore, and acquire a critical understanding of cyber law.
2. Develop competencies for dealing with frauds and deceptions (confidence tricks, scams) and other cyber crimes for example, child pornography etc. That is taking place via the internet.

UNIT 1:

Introduction Computers and its Impact in Society, Overview of Computer and Web Technology, Need for Cyber Law, Cyber Jurisprudence at International and Indian Level, Cyber Law - International Perspectives UN & International Telecommunication Union (ITU) Initiatives Council of Europe - Budapest Convention on Cybercrime, Asia-Pacific Economic Cooperation (APEC), Organization for Economic Co-operation and Development (OECD), World Bank, Commonwealth of Nations.

UNIT 2:

Constitutional & Human Rights Issues in Cyberspace Freedom of Speech and Expression in Cyberspace, Right to Access Cyberspace – Access to Internet, Right to Privacy, Right to Data Protection, Cyber Crimes & Legal Framework Cyber Crimes against Individuals, Institution and State, Hacking, Digital Forgery, Cyber Stalking/Harassment, Cyber Pornography, Identity Theft & Fraud Cyber terrorism, Cyber Defamation.

UNIT 3:

Cyber Torts Cyber Defamation, Different Types of Civil Wrongs under the IT Act 2000, Intellectual Property Issues in Cyber Space Interface with Copyright Law, Interface with Patent Law, Trademarks & Domain Names Related issues

UNIT 4:

E-Commerce Concept, E-commerce-Salient Features, Online approaches like B2B, B2C & C2C Online contracts, Click Wrap Contracts, Applicability of Indian Contract Act, 1872,

UNIT 5:

Dispute Resolution in Cyberspace, Concept of Jurisdiction, Indian Context of Jurisdiction and IT Act, 2000. International Law and Jurisdictional Issues in Cyberspace, Dispute Resolutions.

Books:

1. Chris Reed & John Angel, Computer Law, OUP, New York.
2. Justice Yatindra Singh, Cyber Laws, Universal Law Publishing Co, New Delhi.
3. Verma S, K, Mittal Raman, Legal Dimensions of Cyber Space, Indian Law Institute.
4. Jonathan Rosenoer, Cyber Law, Springer, New York.
5. Sudhir Naib, The Information Technology Act, 2005: A Handbook, OUP, New York.
6. S. R. Bhansali, Information Technology Act, 2000, University Book House Pvt. Ltd.

Course Outcomes:

1. Make Learner Conversant With The Social And Intellectual Property Issues Emerging From ‘Cyberspace.
2. Explore The Legal And Policy Developments In Various Countries To Regulate Cyberspace.
3. Develop The Understanding Of Relationship Between Commerce And Cyberspace.
4. Give Learners In Depth Knowledge Of Information Technology Act and Legal Framework Of Right To Privacy, Data Security And Data Protection.
5. Make Study On Various Case Studies On Real Time Crimes.

2. INTERNET OF THINGS

Course Objectives:

1. To understand Smart Objects and IoT Architectures
2. To learn about various IOT-related protocols
3. To build simple IoT Systems using Arduino and Raspberry Pi.
4. To understand data analytics and cloud in the context of IoT
5. To develop IoT infrastructure for popular applications

Module 1: FUNDAMENTALS OF IoT

Evolution of Internet of Things - Enabling Technologies – IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT models – Simplified IoT Architecture and Core IoT Functional Stack –– Fog, Edge and Cloud in IoT – Functional blocks of an IoT ecosystem – Sensors, Actuators, Smart Objects and Connecting Smart Objects

Module 2: IoT PROTOCOLS

IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN – Network Layer: IP versions, Constrained Nodes and Constrained Networks – Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks – Application Transport Methods: Supervisory Control and Data Acquisition – Application Layer Protocols: CoAP and MQTT

Module 3: DESIGN AND DEVELOPMENT

Design Methodology - Embedded computing logic - Microcontroller, System on Chips - IoT system building blocks - Arduino - Board details, IDE programming - Raspberry Pi - Interfaces and Raspberry Pi with Python Programming.

Module 4: DATA ANALYTICS AND SUPPORTING SERVICES

Structured Vs Unstructured Data and Data in Motion Vs Data in Rest – Role of Machine Learning – No SQL Databases – Hadoop Ecosystem – Apache Kafka, Apache Spark – Edge Streaming Analytics and Network Analytics – Xively Cloud for IoT, Python Web Application Framework – Django – AWS for IoT – System Management with NETCONF-YANG

Course Outcomes:

Upon completion of the course, the student should be able to:

1. Explain the concept of IoT.
2. Analyze various protocols for IoT.
3. Design a PoC of an IoT system using Rasperry Pi/Arduino
4. Apply data analytics and use cloud offerings related to IoT.
5. Analyze applications of IoT in real time scenario

Textbook:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco Press, 2017

References:

1. Arshdeep Bahga, Vijay Madisetti, Internet of Things A hands-on approach‖, Universities Press, 2015
2. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things Key applications and Protocols, Wiley, 2012 (for Unit 2).
3. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle, "From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence", Elsevier, 2014.

4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), Architecting the Internet of Things, Springer, 2011.
5. Michael Margolis, Arduino Cookbook, Recipes to Begin, Expand, and Enhance Your Projects, 2nd Edition, O'Reilly Media, 2011.

3. MULTIMEDIA SYSTEMS

Course Objective:

Adoption of factual knowledge and development of skills needed for independent development of multimedia systems and applications using available hardware and software tools.

Course Contents:

Introduction to Multimedia, DSP Preliminaries: Fundamentals of Signal and Systems, Transformations, IMAGE Representations and Transformations, Elements of Image Compression and CODING: Lossy and Lossless Image Compressions, Fixed-length and Variable-length Coding, Discrete Cosine Transforms and Coding, Wavelet Transform and Coding, Multimedia Standards: Still Image Compression Standards: JPEG and JPEG 2000, Elements of Video Compression System: DPCM, Motion Estimation, Video Compression Standards: Overview, H.261, H.263, H.264, MPEG-1: Specification, continuity and synchronization, MPEG-2: Overview, scalability, Audio Compression: Overview, MPEG Audio Coder

Text Books and References:

1. K. R. Rao, Zoran S. Bojkovic, Dragorad A. Milovanovic, Multimedia Communication Systems: Techniques, Standards, and Networks, Prentice Hall PTR, 2000.
2. Yun Q Shi,, Huifang Sun, Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards, second EDITION, CRC Press, 2008.
3. John G. Proakis and Dimitris K Manolakis, Digital Signal Processing (4th Edition), Prentice Hall, 2006.
4. Iain Richardson, Iain E. G. Richardson, " H.264 and MPEG-4 Video Compression: Video Coding for NEXT Generation Multimedia,"John Willey 2004.
5. Oge Marques, Practical Image and Video Processing Using MATLAB, Wiley-IEEE Press, 2009.
6. Andreas Spanias, Ted Painter, Venkatraman Atti, Audio Signal Processing and Coding, John Wiely, 2007.

Course Outcome:

1. Students will be able to describe the types of media and define multimedia system.
2. Describe the process of digitizing (quantization) of different analog signals (text, graphics, sound and video).
3. Use and apply tools for image processing, video, sound and animation.
4. Apply methodology to develop a multimedia system.
5. Apply acquired knowledge in the field of multimedia in practice and independently continue to expand knowledge in this field.

4. RESEARCH METHODOLOGY

Course Objectives:

1. To familiarize students with the basics of research and the research process.
2. To identify appropriate research topics and define appropriate research problem and parameters
3. To enable students in conducting research work and formulating research synopsis and reports.
4. To enable students to write a research report and thesis

Course Contents:

Module-I: Meaning of Research - Function of Research-Characteristics of Research-Steps involved in Research-Factors which hinder Research-Significance of Research-Research and scientific methods-Research Process-Criteria of good Research-Problems encountered by Researchers-Literature review.

Module-II: Selecting the Research problem-Necessity of defining the problem-Goals and Criteria for identifying problems for research. Perception of Research problem – Techniques involved in defining the problem-Source of problems-Personal consideration.

Module-III: Formulation of Research design-Need for Research design-Features of a good design-Important concepts related to Research design. Different research designs-Basic principles of experimental designs.

Module-IV: Sampling Technique, Sampling, Population, Sampling Frame, Sample, Bias, and Statistical Terms in Sampling: statistic, parameter, Sampling Distribution, Sampling & non-sampling errors, Probability & Non-Probability Sampling, Sample Size Determination.

Module-V: Data Collection Method, Introduction to Primary & Secondary data, Methods of primary data collection, Methods of secondary data collection, Advantages & disadvantages of data collection. Measurement & Scaling Technique, Scales of Measurement, Questionnaire Designing.

Module-VI: Meaning and Technique of interpretation-Precautions in interpretation-Significance of report writing-Different steps in writing a report-Layout of a Research report. Types of report- Mechanics of writing a research report- Precautions for writing a research report- Conclusion.

Module-VII: The Computer and its Role in Research-The Computer and Computer Technology-The Computer System-Important Characteristics-The Binary Number System-Computer Applications-Computers and Researcher.

Module-VIII: Ethical Issues-Ethical Committees-copyright-Intellectual Property rights and patent law–Plagiarism-Citation and Acknowledgement.

Textbook and References

1. C.R. Kothari, Research Methodology: Methods and Techniques, New Age International (P) Limited, 2004
2. Ranjit Kumar, Research Methodology: a step-by-step guide for beginners, SAGE Publications India Pvt Ltd, 2011

Learning Outcomes:

Upon successful completion of the course you are expected to

1. Develop understanding on various kinds of research, objectives of doing research, research process, research designs and sampling.
2. Have basic knowledge on qualitative research techniques
3. Have adequate knowledge on measurement & scaling techniques as well as the quantitative data analysis
4. Have basic awareness of data analysis-and hypothesis testing procedures

5. GENETICS ALGORITHM

Course Objectives:

Students will try to learn:

1. Some of the fundamental principles of genetic algorithms.
2. GA to find solutions to complex problems.
3. Fundamental principles of Evolutionary computation.

The **course** is intended as a fundamental introduction, as well as a survey of the many aspects of **evolutionary algorithms** (EAs), in particular GA, GP,

COURSE CONTENT:

Introduction to Evolutionary Computation: Biological and artificial evolution, evolutionary computation and AI, different historical branches of EC, a simple genetic algorithm.

Search Operators: Crossover, mutation, crossover and mutation rates, Crossover for real-valued representations, mutation for real-valued representations, combinatorial GA,

Selection Schemes: Fitness proportional selection and fitness scaling, ranking, tournament selection, selection pressure and its impact on evolutionary search. Theoretical Analysis of Evolutionary Algorithms: Schema theorems, convergence of the algorithms, computational time complexity of the algorithms, no free lunch theorem.

Search Operators and Representations: Mixing different search operators, adaptive representations.

Niching and Speciation: Fitness sharing, crowding and mating restriction. Constraint Handling: Common techniques, penalty methods, repair methods, Deb's penalty parameter method.

Multiobjective evolutionary optimization: Pareto optimality, multi-objective evolutionary algorithms: MOGA, NSGA-II, etc. Applications of GA in engineering problems, job-shop scheduling and routing problems

References:

1. M. Mitchell, *An introduction to genetic algorithms*, MIT Press, 1996
2. Goldberg D.E.*Genetic Algorithms in Search, Optimization and Machine Learning*. Pearson Education Asia 2002
3. K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley and Sons, 2009.
4. L. D. Davis, *Evolutionary algorithms*, Springer-Verlag, 1999.
5. K. Srinivasa Raju and D. Nagesh Kumar. *Multicriterion Analysis in Engineering and Management.* PHI Learning Pvt. Ltd., New Delhi, India 2010.

Course Outcomes:

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

1. Implement genetic algorithms in finding solution to complex problem
2. Implement evolutionary computation methods to complex problems

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 475P	PROJECT-I	0	0	12	-	90	210	300	3

Details of the project are to be decided by the Department.

DETAILED SYLLABUS (EIGHTH SEMESTER)

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 481	PROGRAM ELECTIVE – VII	3	0	0	70	30	-	100	3

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 482	**PROGRAM ELECTIVE – VIII**	**3**	**0**	**0**	**70**	**30**	**-**	**100**	**3**

PROGRAM ELECTIVE-VII & PROGRAM ELECTIVE-VIII (Any Two)

1. **ADVANCED COMPUTER ARCHITECTURE**
2. **APPROXIMATION ALGORITHMS**
3. **COMPUTATIONAL GEOMETRY**
4. **COMPUTER VISION**
5. **CRYPTOGRAPHY**
6. **MOBILE ROBOTICS**
7. **NATURAL LANGUAGE PROCESSING**
8. **PRINCIPLE OF ROBOTICS**
9. **PRINCIPLE OF PROGRAMMING LANGUAGE**
10. **REINFORCEMENT LEARNING**
11. **SIMULATION AND MODELING**
12. **SOFTWARE TESTING**

1. ADVANCED COMPUTER ARCHITECTURE

Course Objectives

1. To make students know about the Parallelism concepts in Programming
2. To give the students an elaborate idea about the different memory systems and buses.
3. To introduce the advanced processor architectures to the students.
4. To make the students know about the importance of multiprocessor and multicomputers.
5. To study about data flow computer architectures

Course Contents:

Single-threaded execution, traditional microprocessors, DLP, ILP, TLP, memory wall, Parallel programming and performance issues, Shared memory multiprocessors, Synchronization, small-scale symmetric multiprocessors on a snoopy bus, cache coherence on snoopy buses, Scalable multiprocessors, Directory-based cache coherence, Interconnection network, Memory consistency models, Software distributed shared memory, multithreading in hardware, Chip multiprocessing, Current research and future trends.

Text Books and References

1. K. Hwang and F. A. Briggs *Computer Architecture and Parallel Processing*, McGraw Hill, 1985.
2. H. Stone *Advanced Computer Architecture*, Addison Wesley, 1989.
3. H. J. Siegel *Interconnection Network for Large Scale Parallel Processing*, McGraw Hill, 1990.
4. J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd Edition, Morgan Kaufmann, 1995.

Course Outcomes

At the end of this course student will be able to:

1. Demonstrate concepts of parallelism in hardware/software.
2. Discuss memory organization and mapping techniques.
3. Describe architectural features of advanced processors.
4. Interpret performance of different pipelined processors.
5. Explain data flow in arithmetic algorithms
6. Development of software to solve computationally intensive problems

2. APPROXIMATION ALGORITHMS

Course Objective

1. To provide a solid background on the pertinent computer science, mathematical, and electrical engineering concepts that make up the foundations of the discipline of electrical engineering and computer science engineering, as well as their closely associated fields.-
2. To provide students with the knowledge to correctly apply the laws of nature to the creative formulation and solution of engineering problems through the use of analytical, computational and experimental techniques.-
3. To expand the research of electrical engineering and computer science to non-traditional areas by continually seeking to incorporate new methodologies and research findings to our graduate curriculum

Course Contents:

Review of basic concepts: LP duality, complementary slackness, convex optimization via separation oracles, NP-completeness

Greedy algorithms: Vertex/Set Cover, Submodular Function Maximization Application: Maximizing the spread of influence through a social network

Tree based approximations: Undirected Traveling Salesman Problem and the Held-Karp bounds Undirected Minimum Steiner Tree-Embedding metric spaces into trees-Application: Buy-At-Bulk Network Design-Application: Group Steiner Trees

Simultaneous Optimization: Fairness in load balancing and resource allocation-Data aggregation in sensor networks-Shallow Light Trees

Dynamic programming techniques: Zero-One Knapsack and Bin Packing-Application: Constrained Shortest Paths-Directed Steiner Trees or Geometric PTASs (polynomial time approximation schemes)

Geometric Embeddings: Bourgain's embeddings of metric spaces into normed spaces-The Sparsest Cut Problem, with applications-Semidefinite Programming and Max-Cut-The l1-l2 gap, with algorithmic applications, Open problem: embedding minor-closed graphs

Scheduling Problems and Resource Augmentation: Flow-time with and without augmentation Non-clairvoyant scheduling

Primal-Dual algorithms: Facility Location and the k-Median Problem, Steiner Network Design

Text Books and References

1. Vijay V. Vazirani, Approximation Algorithms, Springer 2003
2. Ding-Zhu Du, Ker-I Ko, Xiaodong Hu, Design and Analysis of Approximation Algorithms, Springer.
3. David P. Williamson, David B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press.

Course Outcome

At the end of the course, students are

1. able to identify novel and significant open research questions in computer science and engineering and are able to situate such questions in the contexts of current research literature.
2. are able to apply their knowledge of computing, mathematics, science, and engineering to the analysis of technological problems, as well as to the design and implementation of viable solutions to those problems
3. possess the characteristics of lifelong learners; they are able to acquire and use new techniques, skills, and engineering and scientific tools for research and development in electrical engineering and computer science, as well as to develop new methods and make new discoveries
4. communicate effectively through oral, visual, and written means, effectively addressing a broad range of technical audiences

3. COMPUTATIONAL GEOMETRY

Course Objective

1. Provide essentials of Computational Geometry. In-depth study of the fundamental geometric structures and techniques used in this field
2. Advanced geometric algorithms, to deal with the large number of real world problems from other fields, such as Wireless and Mobile Computing, Computer Graphics, Computer Vision, Databases, Robotics, and VLSI design.

Algorithmic design paradigms (divide and conquer, incremental, sweep line, and prune and search) and basic data structures (segment and interval trees). Geometric searching: point locations (slab and chain methods) and range searching (kD and range trees); Convex hull: Graham's scan, gift wrapping, quick hull, divide-and-conquer; Voronoi diagram and Delaunay triangulation: properties and construction algorithms (sweep line and divide-and-conquer algorithms). Visibility and Art gallery problems, motion PLANNING and shortest paths. Arrangements and duality; Line segments intersection problem; closest pair computation.

Text Books and References

1. F. P. Preparata And M. I. Shamos, Computational Geometry: An Introduction, Springer-Verlag, 1985.
2. J. O'rourke, Computational Geometry In C, 2nd Ed, Cambridge University Press, 1998.
3. M. Laszlo, Computational Geometry And Computer Graphics In C++, Prentice-Hall, 1996.
4. M. De Berg, M. Van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms And Applications, Springer -Verlag, 1997

Course Outcome

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

1. Design of efficient techniques for the computer-based representation and manipulation of geometric objects.
2. Understand the complexity of geometric problems

4. COMPUTER VISION

Course Objective

1. To discuss students ideas of image formation, major ideas, methods and techniques of computer vision and pattern recognition;
2. To develop appreciation for various issues in the design of computer vision and object recognition systems and finally to provide students the vast programming experience in computer vision and object recognition application.

Course Contents:
Digital Image Formation and low-level processing Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Depth estimation and Multi-camera views Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Feature Extraction Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

Image Segmentation Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Pattern Analysis Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

Motion Analysis Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Shape from X Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

Miscellaneous Applications: CBIR, CBVR, Activity Recognition, computational photography, Biometrics, stitching and document processing; Modern trends - super-resolution; GPU, Augmented Reality; cognitive models, fusion and SR&CS.

Textbooks and References:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Pearson Education, 2003.
3. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
4. K. Fukunaga, Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.
5. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992

Course Outcome
After the course, the students will be able
1. To identify the basic concepts, terminology, theories. ,models and methods of computer vision and finally able to design for computer vision for specific applications.

5. CRYPTOGRAPHY

Course Objective

1. To understand the fundamentals of cryptography and its types.
2. To understand network security threats, security services, and their countermeasures.
3. To understand RSA algorithms.
4. To acquire background on well known network security protocols such SSL/TLS.
5. To understand hash functions; authentication; firewalls.

UNIT I : Security trends – Attacks and services – Classical cryptosystems – Different types of ciphers – LFSR sequences – Basic Number theory – Congruences – Chinese Remainder theorem – Modular exponentiation – Fermat and Euler's theorem – Legendre and Jacobi symbols – Finite fields – continued fractions.

UNIT II: Simple DES – Differential cryptanalysis – DES – Modes of operation – Triple DES –AES – RC4 – RSA – Attacks – Primality test – factoring.

UNIT III: Discrete Logarithms – Computing discrete logs – Diffie-Hellman key exchange –ElGamal Public key cryptosystems – Hash functions – Secure Hash – Birthday attacks -MD5 – Digital signatures – RSA – ElGamal – DSA.

UNIT IV: Authentication applications – Kerberos, X.509, PKI – Electronic Mail security – PGP, S/MIME – IP security – Web Security – SSL, TLS, SET.

UNIT V: System security – Intruders – Malicious software – viruses – Firewalls – Security Standards.

Text Books and References

1. William Stallings, "Cryptography and Network Security", Pearson/PHI, 4th ed, 2006.
2. Kahate , Cryptography And Network Security Paperback,2007,TMH.

Course Outcome

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

1. Understand various Cryptographic Techniques
2. Apply various public key cryptography techniques
3. Implement Hashing and Digital Signature techniques
4. Understand the various Security Applications

MOBILE ROBOTICS

Course Objective:

The Student should be made to:

1. Discuss about the need and application of robotics in our society.
2. To address different types of robots and their functionality in detailed.

Course Contents:

Introduction to Mobile robot architectures, Control Paradigms, Sensors and actuators. Learning Approaches for robots. Navigation Strategies, Detecting and handling Novelty. Behavior-based robotics, AlE and their APPLICATION to robots. Case studies of learning robots, Laboratory sessions will include study and implementations of the above methodologies using real robots.

Books and References

1. U. Nehmzow, Mobile Robotics - A Practical Introduction, 2nd Ed, Springer, 2003.
2. L. N. de Castro and J. Timmis, Artificial Immune SYSTEMS: A New Computational Intelligence Approach, Springer, 2002.
3. D. Dasgupta, Artificial Immune Systems and Their APPLICATIONS, Springer, 1999.
4. R. C. Arkin, Behaviour Based Robotics, MIT PRESS, 1998.

Course Outcomes:

Upon completion of the course, students will be able to

1. Students will learn robot programming and how to make a small robot for navigation.
2. Student will learn how to develop algorithm for putting intelligence into robot

6. NATURAL LANGUAGE PROCESSING

Course Objectives:

1. Teach students the leading trends and systems in natural language processing.
2. Make them understand the concepts of morphology, syntax, semantics and pragmatics of the language.
3. Teach them to recognize the significance of pragmatics for natural language understanding.
4. Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic and semantic.

Module-I: Introduction to Natural Language Understanding: The study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different Levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English syntax.

Module-II: Introduction to semantics and knowledge representation, Some applications like machine translation, database interface.

Module-III: Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing. Feature Systems and Augmented Grammars: Basic Feature system for English, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.

Module-IV: Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser.

Module-V: Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Part-of-Speech tagging, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Best First Parsing. Semantics and Logical Form, Word senses and Ambiguity, Encoding Ambiguity in Logical Form.

Suggested Text Book & References:

1. Daniel Jurafsky and James H Martin. Speech and Language Processing, 2e, Pearson Education, 2009
2. James A. Natural language Understanding 2e, Pearson Education, 1994.
3. Bharati A., Sangal R., Chaitanya V. Natural language processing: a Paninian Perspective, PHI, 2000.
4. Siddiqui T., Tiwary U. S. Natural language processing and Information retrieval, OUP, 2008.

Course Outcomes:

After successful completion of this course, students will be able to

1. Understand approaches to syntax and semantics in NLP.
2. Understand approaches to discourse, generation, dialogue and summarization within NLP.
3. Understand current methods for statistical approaches to machine translation.
4. Understand machine learning techniques used in NLP, including hidden Markov models and probabilistic context-free grammars, clustering and unsupervised methods.

7. PRINCIPLE OF ROBOTICS

Course Objective:-

The objectives of the course are to enable the students to

1. Understand the components and their working principles of a robotic system.
2. Learn forward kinematics, inverse kinematics and dynamic modeling of manipulators.
3. Learn robot programming and industrial application of robots.

UNIT I: BASIC CONCEPTS

Brief history-Types of Robot–Technology-Robot classifications and specifications-Design and control issues- Various manipulators – Sensors – work cell – Programming languages.

UNIT II: DIRECT AND INVERSE KINEMATICS

Mathematical representation of Robots - Position and orientation - Homogeneous transformation-Various joints- Representation using the Denavit Hattenberg parameters -Degrees of freedom-Direct kinematics-Inverse kinematics-PUMA560 & SCARA robots- Solvability – Solution methods-Closed form solution.

UNIT III: MANIPULATOR DIFFERENTIAL MOTION AND STATICS

Linear and angular velocities-Manipulator Jacobian-Prismatic and rotary joints–Inverse - Wrist and arm singularity – Static analysis – Force and moment Balance.

UNIT IV: PATH PLANNING

Definition-Joint space technique-Use of p-degree polynomial-Cubic polynomial-Cartesian space technique – Parametric descriptions – Straight line and circular paths – Position and orientation planning.

UNIT V: DYNAMICS AND CONTROL

Lagrangian mechanics-2 DOF Manipulator-Lagrange Euler formulation-Dynamic model - Manipulator control problem-Linear control schemes-PID control scheme-Force control of robotic manipulator.

Text Books and References

1. Ashitava Ghoshal, Robotics-Fundamental Concepts and Analysis', Oxford University Press, Sixth impression, 2010.
2. K. K. Appu Kuttan, Robotics, I K International, 2007.
3. Edwin Wise, Applied Robotics, Cengage Learning, 2003.
4. R.D. Klafter, T.A. Chimielewski and M. Negin, Robotic Engineering–An Integrated Approach Prentice Hall of India, New Delhi, 1994.
5. B.K. Ghosh, Control in Robotics and Automation: Sensor Based Integration, Allied Publishers, Chennai, 1998.
6. S. Ghoshal, "Embedded Systems & Robotics"–Projects using the 8051 Microcontroller", Cengage Learning, 2009.

Course Outcomes:

Upon completion of the course, students will be able to understand:

1. Understand about the basics of automation and robotics
2. Explain the basics of Control and analysis of robotics motion
3. Principles of robot programming and handle with typical robot
4. Working of mobile robots
5. Explain about the applications of robots

8. PRINCIPLE OF PROGRAMMING LANGUAGE

Course Objective:

1. To know the specification of a programming language.
2. To know the types of grammars used in a programming language
3. To know Procedural and OO Languages
4. To understand Functional Programming and Logic Programming
5. To learn various Programming Language Constructs

Overview of programming languages; language design and implementation - Grammars (regular expressions, context-free grammars); parse trees; recursive descent parsing, interpretation, and compilation -Expressions; control structures; subroutines -Storage management; scoping rules; bindings for names - Principles of object-oriented languages (data abstraction, encapsulation, inheritance, polymorphism); implementation mechanisms - Functional programming principles; Scheme -Data types - Other topics.

Books and References

1. Robert W. Sebesta, Principles of Programming Languages, Pearson Education, 2012.
2. Seema Kedar, Principles of Programming Languages, Technical Publication, 2011.
3. Dr. Sachin Kumar (Author), Kadambri Agarwal, Principles of Programming Languages, S.K. Kataria & Sons, 2013.
4. Principles of Programming Languages, Professional Publications, 2014.
5. Kenneth C. Louden, Programming Languages: Principles and Practice, Cengage Learning, 2002

Course Outcome:

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

1. Know the types of grammars used in programming languages.
2. Understand different language features used in current programming languages.
3. Analyze to program in different language paradigms and evaluate their relative benefits.
4. An understanding of the key concepts in the implementation of common features of programming languages.

9. REINFORCEMENT LEARNING

Course Objective:

1. This course introduces you to statistical learning techniques where an agent explicitly takes actions and interacts with the world.
2. It will also help us to know the general purpose formalism for automated decision-making and AI.

The Reinforcement Learning problem: evaluative feedback, non-associative learning, Rewards and RETURNS, Markov Decision Processes, Value functions, optimality and approximation

Dynamic programming: value iteration, policy iteration, asynchronous DP, generalized policy iteration

Monte-Carlo methods: policy evaluation, roll outs, on policy and off policy learning, importance sampling

Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, R-learning, Games and after states

Eligibility traces: n-step TD prediction, TD(lambda), forward and backward views, Q(lambda), SARSA(lambda), replacing traces and accumulating traces

Function Approximation: Value prediction, gradient descent methods, linear function approximation, ANN based function approximation, lazy learning, instability issues

Policy Gradient methods: non-associative learning - REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods

Planning and Learning: Model based learning and planning, prioritized sweeping, Dyna, heuristic search, trajectory sampling, E^3 algorithm

Hierarchical RL: MAXQ framework, Options framework, HAM framework, airport algorithm, hierarchical policy gradient

CASE studies: Elevator dispatching, Samuel's checker player, TD-gammon, Acrobot, Helicopter piloting

Text Books and References

1. R. S. Sutton and A. G. Barto: "Reinforcement Learning: An Introduction", Cambridge, MA: MIT Press, 1998.
2. "Neuro-dynamic programming". Dimitri P. Bertsikas and John N. Tsitsiklis
3. "Learning Automata - An Introduction". Kumpati S. Narendra and M. A. L. Thathachar

Course Outcome:

1. Understand basic exploration methods and the exploration/exploitation tradeoff
2. Understand value functions, as a general-purpose tool for optimal decision-making
3. Know how to implement dynamic programming as an efficient solution approach to an industrial control problem
4. This course teaches you the key concepts of Reinforcement Learning, underlying classic and modern algorithms in RL.
5. After completing this course, you will be able to start using RL for real problems, where you have or can specify the MDP.

10. SIMULATION AND MODELLING

Course Objective:

1. Define the basics of simulation modelling and replicating the practical situations in organizations
2. Generate random numbers and random variates using different techniques.
3. Develop a simulation model using heuristic methods.
4. Analysis of Simulation models using input analyzer, and output analyser
5. Explain Verification and Validation of simulation model.

Introduction to Probability theory, Random variables, commonly used continuous and discrete distributions. Introduction to Stochastic Process, Poisson process, Markov chains, steady state and transient analysis. Pseudo random numbers: Methods of Generation and testing. Methods for generating continuous and discrete distributions. Methods for generating Poisson Process. Building blocks of Simulation, Data Structures and Algorithms. Introduction to Probabilistic modelling, Maximum Likelihood Variance reduction techniques: antithetic variates, control variates, common random numbers, importance sampling. Analysis of Simulation results: confidence intervals, design of experiments. Markov Chain Monte Carlo techniques.

Books and References:

1. Sheldon M. Ross: Introduction to Probability Models 7th Edition, Academic Press, 2002.
2. Donald E. Knuth: The Art of Computer Programming - Volume 2: Semi Numerical Algorithms, 2nd Edition, Addison Wesley, Reading MA, USA 2000.
3. Sheldon M. Ross Simulation 3rd Edition, Academic Press, 2002
4. M. Law and W. D. Kelton: Simulation Modeling and Analysis, 3rd Edition, McGraw Hill, New York, USA, 1998
5. Raj Jain The Art of Computer Systems Performance Analysis, John Wiley and Sons, New York, USA, 1991

Course Outcomes:

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

1. Describe the role of important elements of discrete event simulation and modelling paradigm
2. Conceptualize real world situations related to systems development decisions, originating from source requirements and goals.
3. Develop skills to apply simulation software to construct and execute goal-driven system models
4. Interpret the model and apply the results to resolve critical issues in a real world environment

11. SOFTWARE TESTING

Course Objectives:

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to plan a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
5. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
6. To understand software test automation problems and solutions.
7. To learn how to write software testing documents, and communicate with engineers in various forms.
8. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Unit-I: Introduction

Faults, Errors, and Failures, Basics of software testing, Testing objectives, Principles of testing, Requirements, behavior and correctness, Testing and debugging, Test metrics and measurements,

Verification, Validation and Testing, Types of testing, Software Quality and Reliability, Software defect tracking.

Unit-II: White Box and Black Box Testing

White box testing, static testing, static analysis tools, Structural testing: Unit/Code functional testing, Code coverage testing, Code complexity testing, Black Box testing, Requirements based testing, Boundary value analysis, Equivalence partitioning, state/graph based testing, Model based testing and model checking, Differences between white box and Black box testing.

Unit-III: Integration, System, and Acceptance Testing
Top down and Bottom up integration, Bi-directional integration, System integration, Scenario Testing, Defect Bash, Functional versus Non-functional testing, Design/Architecture verification, Deployment testing, Beta testing, Scalability testing, Reliability testing, Stress testing, Acceptance testing: Acceptance criteria, test cases selection and execute

Unit-IV: Test Selection & Minimization for Regression Testing
Regression testing, Regression test process, Initial Smoke or Sanity test, Selection of regression tests, Execution Trace, Dynamic Slicing, Test Minimization, Tools for regression testing, Ad hoc Testing: Pair testing, Exploratory testing, Iterative testing, Defect seeding.

Unit-V: Test Management and Automation
Test Planning, Management, Execution and Reporting, Software Test Automation: Scope of automation, Design & Architecture for automation, Generic requirements for test tool framework, Test tool selection, Testing in Object Oriented Systems.

Books and References:

1. S. Desikan and G. Ramesh, "Software Testing: Principles and Practices", Pearson Education.
2. Aditya P. Mathur, "Fundamentals of Software Testing", Pearson Education
3. Naik and Tripathy, "Software Testing and Quality Assurance", Wiley
4. K. K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International Publication.

Course Outcome:
By the end of the course, Student should:
1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability to understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems
7. Have an ability to use software testing methods and modern software testing tools for their testing projects.

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 483	OPEN ELECTIVE-III	3	0	0	70	30	-	100	3

OPEN ELECTIVE-III

1. **DIGITAL SIGNAL PROCESSING**
2. **WIRELESS SENSOR NETWORKS**
3. **DEEP LEARNING**
4. **DIGITAL IMAGE PROCESSING**

1. DIGITAL SIGNAL PROCESSING

Course Objective:

Students will try to learn:

1. Introduce the concepts of digital signal processing and the basic analytical methods and show how they are applied to design filters for given applications.
2. To develop skills for analyzing and synthesizing algorithms and systems that process discrete time signals, with emphasis on realization and implementation.
3. To explore the properties of DFT in mathematical problem solving.
4. To illustrate FFT calculations mathematically and develop FFT based DSP algorithms.
5. To introduce DSP processor for real time signal processing application
6. To develop skills to design finite impulse response (FIR) filters to satisfy a desired frequency response.

Discrete-Time Signals and Systems: Discrete Time Signals, Analysis of Discrete Time Linear Time Invariant Systems,Systems described by Difference Equation, Correlation of Discrete Time Signals. Design of Digital Filters: Digital Filters by placement of poles and zeros in the Z-plane. Low pass, High pass and band pass Filters, Notch Filter; Comb Filter & All pass Filter. Realization of FIR & IIR systems, Design of FIR Filter using windows, Design of IIR filter by the Bilinear Transformation method. Discrete Fourier Transform : DFT and its relationship to other transforms, properties of the DFT, Circular convolution in time and frequency domains, Linear convolution in time and frequency domain by overlap save and overlap add methods. Fast Fourier Transform: Adaptive Filter, Inverse system, Deconvolution and System Identification. Power Spectrum Estimation: Estimation of Autocorrelation and power spectrum of random signals, use of DFT in Power Spectrum Estimation. Parametric method - The Burg Method for the AR Model Parameters, Least Square Method, ARMA Model for Power Spectrum Estimation. The Adaptive Linear Combiner, Wiener Filters, Adaptive Transversal Filter Using Gradient Vector Estimation, LMS algorithm and its convergence analysis.

Books and References

1. J. G. Proakais and D. G. Manolakis, Digital Signal Processing: Principles Algorithms and Applications, Pearson Education, 2005.
2. A. V. Oppeinheim and R. W. Schafer, Digital Signal Processing, Pearson Education, 2004.
3. S. K. Mitra, Digital Signal Processing: A computer based approach, TMH, 2001.
4. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Pearson Education, 2004.

Course Outcomes:

Students will be able to:

1. To understand the concept of DT Signal and perform signal manipulation
2. To perform analysis of DT system in time domain
3. To develop FFT flow-graph and Fast DSP Algorithms.
4. To design DSP system for Real Time Signal Processing.
5. Understand the process of converting the continuous-time signal into digital signal, process it and convert back to continuous-time signal.
6. Apply the tools like DFT and z-transform to analyze and design the digital LTI systems.
7. Apply the digital filter structures and DSP processor knowledge to implement the actual digital LTI systems and evaluate the effect of finite word-length effects.

2. WIRELESS SENSOR NETWORKS

Course Objectives:

1. To understand the basics of WSNs.
2. To learn architecture and placement strategies of Sensors.
3. To learn various clustering and routing algorithms in WSNs.
4. To learn and develop sensor algorithms.

Module-I: Introduction to wireless sensor Networks - Advantages of ad-hoc/sensor networks- Unique constraints and challenges.

Module-II: Sensor node hardware- mica2- micaZ- telosB- Imote2- Sensor node OS-tinyoS-Programming tools-C-Java.

Module-III: WSN coverage and placement- coverage Algorithm- Placement Problem-Topology management in wireless sensor Networks- Different classification of topology management Algorithms- Medium access control in wireless networks

Module-III: Routing in sensor networks- position based routing- Clustered based routing Algorithms.

Module-IV: Congestion and flow control- Source of congestion- congestion control scenarios- Protocols for congestion and flow control in sensor networks.

Module-V: Hardware design of sensor Networks-Characteristics–Design challenges-Design of Architecture- Functional components- Energy supply- operating system.

Module-VI: Application-Underwater sensor networks-Real life deployment of WSNs-development of sensor based networking for improved management of irrigated crops - usage of sensors in military.

Course Outcome:

1. To understand the basis of sensors with its applications
2. To learn the architecture of Sensors.
3. To analyze clustering and routing algorithms
4. To design and develop sensor algorithms (clustering and routing algorithms etc.)

3. DEEP LEARNING

Course Objectives

1. Understand the context of deep learning
2. Understand the data needs of deep learning
3. Have a working knowledge of deep learning

Course Content

Introduction to Deep Learning: Bayesian Learning, Linear Classifiers, Linear Machines with Hinge Loss, Gradient Descent, Batch Optimization, Unsupervised Learning with Deep Network, Autoencoders, Convolutional Neural Network, Building blocks of CNN, Transfer Learning, Revisiting Gradient Descent, Momentum Optimizer, Effective training in Deep Net- early stopping, Dropout, Batch Normalization. Need of RNN: RNN network, limitations of RNN, Introduction to LSTM Networks and GRU network.

Text Books and Reference Books:

1. Charu C. Aggarwal "Neural Networks and Deep learning" Springer International Publishing, 2018
2. Satish Kumar, "Neural Networks, A Classroom Approach", Tata McGraw -Hill, 2007.
3. Simon Haykin, "Neural Networks, A Comprehensive Foundation", 2nd Edition, Addison Wesley Longman, 2001.
4. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006
5. Duda, Richard, Peter Hart, and David Stork. Pattern Classification. 2nd ed. New York, NY: Wiley-Interscience, 2000.

Course Outcome:

1. Understand the basic concepts in Deep Learning and applications
2. Understand various Deep Learning Algorithms

4. DIGITAL IMAGE PROCESSING

Course Objective:

1. Describe and explain basic principles of digital image processing.
2. Design and implement algorithms that perform basic image processing (e.g. noise removal and image enhancement).
3. Design and implement algorithms for advanced image analysis (e.g. image compression, image segmentation).
4. Assess the performance of image processing algorithms and systems.

Course content:

Elements of Visual perception. Image sensing and Acquisition . Imaging in different bands. Digital Image Representation. Relationship between pixels. Image transformations: 2D-DFT, DCT, DST, Hadamard, Walsh, Hotelling transformation, 2D-Wavelet transformation, Wavelet packets.

Image Enhancements in spatial domain and Frequency domain. Image Restoration techniques. Color Image processing.

Error free compression: Variable length coding, LZW, Bit-plane coding, Lossless predictive coding Lossy compression: Lossy predictive coding, transform coding, wavelet coding. Image compression standards, CCITT, JPEG, JPEG 2000, Video compression standards.

Summary of morphological operations in Binary and Gray Images. Image segmentation: Point, Line and Edge segmentation. Edge linking and Boundary detection. Segmentation using thresholding, Region based segmentation. Segmentation by morphological watersheds. Use of motion in segmentation.

Feature Extraction from the Image: Boundary descriptors, Regional descriptors, Relational descriptors.

Text Books:

1. R. C. Gonzalez, R.E. Woods," Digital Image processing", Pearson edition, Inc3/e, 2008.
2. A.K. Jain," Fundamentals of Digital Image Processing", PHI,1995

Reference Books:

1. J.C. Russ," The Image Processing Handbook", (5/e), CRC, 2006
2. R.C.Gonzalez & R.E. Woods; "Digital Image Processing with MATLAB", Prentice Hall, 2003

Course Outcomes:

After the completion of the course, the students will be able to

1. Analyze general terminology of digital image processing.
2. Examine various types of images, intensity transformations and spatial filtering.
3. Develop Fourier transform for image processing in frequency domain.
4. Evaluate the methodologies for image segmentation, restoration etc.
5. Implement image process and analysis algorithms.
6. Apply image processing algorithms in practical applications.

PRACTICAL/DESIGN

Course No.	Subject	Hours/Week			Marks				Credit
		L	T	P	Theory	Sess.	Pract.	Total	
CS 484P	PROJECT-II	0	0	16	-	120	280	400	8

In continuation and fulfillment of the CS 475P.

Total Credits from 1st to 8th Sem. = 17.5+21+23+23.5+23+22+16+17 = 163

Total Marks from 1st to 8th Sem. = 500+650+800+850+900+900+800+700 = 6100