SECTION – 2

DESCRIPTION OF COURSES OFFERED

2.0 Schema of B.Tech. First Year (Common) Course

B.TECH. SEMESTER - I		
Sub. Code	Sub. Name	Credit
MA 1101	Engineering Mathematics –I	4
CE 1102	Mechanics of Solids	4
PH 1103	Engineering Physics	4
ME 1104	Elements of Mechanical Engineering	4
ME 1105	Engineering Graphics	3
BA 1106	Communication Skills	3
BA 1165	Universal Human Values & Professional Ethics, Lab - I	1
ME 1161	Workshop Practice	1.5
PH 1162	Engineering Physics Lab	1.5
	Total credits for the Semester:	26

B.TECH. SEMESTER - II		
Sub. Code	Sub. Name	Credit
MA 1201	Engineering Mathematics –II	4
EC 1107	Basic Electronics	4
CH 1108	Engineering Chemistry	4
EE 1109	Elements of Electrical Engineering	4
CS 1110	Computer Programming using C	4
CH 1111	Environmental Science	2
BA 1261	Universal Human Values & Professional Ethics, Lab - II	1
CH 1163	Engineering Chemistry Lab	1.5
CS 1164	Computer Programming Lab	1.5
	Total credits for the Semester:	26

2.1 Detailed Short Syllabus of B.Tech. First Year (Common) Course

B.TECH. – SEMESTER – I

MA 1101: ENGINEERING MATHEMATICS –I, Credit: 4 (L-3, T-1, P-0)

Successive differentiation, Leibnitz's theorem, Polar curves, Tangent and normal of Polar curves, Angle between radius vector and tangent, Angle of intersection of two curves, Derivatives of arcs (Cartesian and Polar), Asymptotes, Curvature, Rolle's theorem, Mean value theorems, Expansion of series, Partial differentiation, Total differential, Differentiation of composite and implicit functions. Tracing of curves, Integral calculus, Analytical solid geometry- Direction Cosines, Planes, Straight lines, Spheres, Right circular cone and Right circular cylinder; **Infinite series-** Convergence, Divergence, Comparison test, Ratio test, Raabe's test, Cauchy's root test, Cauchy's integral test, Alternating series, Leibnitz's test, Absolute and conditional convergence.

CE 1102: MECHANICSOF SOLIDS, Credit: 4 (L-3, T-1, P-0)

Coplanar Concurrent Force System, Coplanar Non-Concurrent Force System, Centroids and Second Moment of Areas, Simple Stresses and Strains, Kinematics, Kinetics – Application of Newton's Second Law, Application of Work-Energy Principle, Application Impulse -Momentum principle.

PH 1103: ENGINEERING PHYSICS, Credit: 4 (L-3, T-1, P-0)

Vibrations, Oscillators, Resonance, Waves, Interference of light waves, Young's experiment, Thin film interference, Newton's ring, Diffraction of light, Fraunhofer diffraction and plane transmission grating, Rayleigh criterion, Polarization, Double refraction, Plane, Circularly and elliptically polarized light, Inadequacy of classical mechanics, Black body radiation, Rayleigh Jeans' law, Wien's displacement law, Planck's radiation law, Planck's quantum hypothesis, Photoelectric effect, Wave particle duality, de Broglie waves, Matter waves (Davisson-Germer experiment), Group velocity and phase velocity, Wave packets and Heisenberg's uncertainty principle, Wave function and its physical significance, Schrodinger's equation, Schrodinger's 1-D time independent equations, Potential well, potential barrier and quantum tunneling. Concept of free electron theory, Quantum theory of free electrons, Fermi energy, Effect of temperature in Fermi-Dirac distribution, Bloch theorem, Concept of energy levels and bands, Distinction between Insulator, Semi conductors and Conductors in terms of energy band, p-n junction. Lecture(s) on recent trends in Physics in engineering perspective (Non-credit).

ME 1104 ELEMENTS OF MECHANICAL ENGINEERING, Credit: 4 (L-3, T-1, P-0)

Thermodynamics: Introduction, Thermodynamic processes, First law of thermodynamics, Second law of thermodynamics, Properties of gases, Air compressor, Internal Combustion Engines, Fluid Mechanics: Fluid Properties, Fluid statics, Fluid kinematics and Fluid dynamics. Pump: Centrifugal and Reciprocating Pump. Transmission of Motion

and Power: Belt and Gear Drive. Welding, Metal Cutting and Machine Tools, Lathe, Drilling Machine, Grinding Machine.

ME 1105 ENGINEERING GRAPHICS, Credit: 3 (L-3, T-0, P-0)
Lettering, conventions and dimensioning, Scales, Projection of points, Projections of lines, Projection of planes, Projection of solids, Section of solids, Development of surfaces, Orthographic Projection, Isometric Projection.

BA 1106: COMMUNICATION SKILLS, Credit: 3 (L-3, T-0, P-0)
Introduction and Understanding Communication Skills, 7 C's of Communication, Verbal Communication- 3 V's of Communication, Non Verbal Communication, Essay Writing, Expansion of idea, Comprehension, Vocabulary, Report Writing, Business Correspondence, E-mail Writing. Grammar, Class Room Practice / Language Lab (Not to be included in Question Paper), Oral Communication, Extempore, Group Discussion, Power Point Presentation, Role Play.

BA 1165: UNIVERSAL HUMAN VALUES & PROFESSIONAL ETHICS, LAB – I, Credit: 1
Introduction, Need, Basic Guidelines, Content and Process for Value Education, Understanding Harmony in the Human Being- Harmony in Myself!, Understanding Harmony in the Family - Harmony in Human- Human Relationship.

ME 1161: WORKSHOP PRACTICE, Credit: 1.5 (L-0, T-0, P-3)
Demonstration of Carpentry, Fitting, Plumbing and Soldering. 12 labs are to be conducted in a semester.

PH 1162: ENGINEERING PHYSICS LAB, Credit: 1.5 (L-0, T-0, P-3)
12 labs are to be conducted on the basis of the syllabus of the corresponding theory paper.

B.TECH. – SEMESTER – II

MA 1201: ENGINEERING MATHEMATICS, Credit: 4 (L-3, T-1, P-0)
Ordinary differential equations (ODE) – I, Ordinary differential equations (ODE)- II, Laplace transforms, Vector spaces and subspaces, Simple examples. Matrices, Solution by Gauss elimination. Taylor's theorem for a function of two variables, Extreme values of a function of two variables, Lagrange's method of undetermined multipliers, Multiple integrals, Beta and Gamma functions.

EC 1107: BASIC ELECTRONICS, Credit: 4 (L-3, T-1, P-0)
Characteristics, types and applications of resistors, Capacitors and inductors, Diode characteristics and their applications, Special purpose diode characteristics and their applications, : Characteristics of transistors, Transistor biasing, stabilization and applications, Introduction to digital electronics, Introduction to communication system and Internet.

CH 1108: ENGINEERING CHEMISTRY, Credit: 4 (L-3, T-1, P-0)
Electrochemistry - Electrochemical changes, measurement of electrode potential, Nernst equation and its applications, electrochemical series & its applications, electrochemical cell and its classifications, types of electrodes, electromotive force, standard cell, determination of EMF, concentration cell, liquid junction potential, salt bridge. Battery - The lead-acid storage cell, lithium-ion battery, H2–O2 fuel cell. Corrosion and its control – introduction, types and mechanism of different types of corrosion, passivity, galvanic series, factors, corrosion control - corrosion inhibitors, cathodic protection - sacrificial anodic and impressed current cathodic protection. Electroplating and Painting. Principle of Organic Chemistry – Electron displacement effects – types, inductive effect and its applications, mesomeric effects and its applications. Aromaticity – Introduction, induced ring current, characteristics and theoretical criteria, examples, anti-aromaticity. Liquid crystals - Introduction, classification, molecular ordering of thermotropic & lyotropic liquid crystal, chemical constitution, homologous series, applications. Polymer - Definition, polymerization, natural rubber, tacticity, conducting polymers, (LDPE) and (HDPE), molecular weight analysis. Nano Science - Introduction, applications, fullerenes and carbon nanotubes.

EE 1109: ELEMENTS OF ELECTRICAL ENGINEERING, Credit: 4 (L-3, T-1, P-0)

DC Circuits, Network Reduction Technique, Magnetic Circuits, Single Phase AC Circuits, Three Phase AC Circuits: Symmetrical sinusoidal supply systems, voltage, current and power relationship in 3-phase balanced star and delta connected loads, Two wattmeter method of measurement of power, Transformers, Three phase induction motor, power system.

CS 1110: COMPUTER PROGRAMMING USING C, Credit: 4 (L-3, T-1, P-0)

Introduction to Computer Fundamentals & Programming Language, Constants, variables, and data types, Operators and expressions, Decision making and branching & looping, Arrays (One Dimensional Array, Two Dimensional Array, Strings), User defined functions & Macro, Pointers, Structures and Unions, File management in C, Dynamic memory Allocation, Linked List.

CH 1111: ENVIRONMENTAL SCIENCE, Credit: 2 (L-2, T-1, P-0)

Fundamentals of Environmental Science, Ecology, Mass Awareness of Chemical Disarmament, Chemical weapons convention (CWC) and awareness of advanced technological applications for the welfare of the entire humankind. Peaceful uses of chemistry, application of green chemistry in pollution control and Geoscience. Statistics in Environmental Science, populations, samples, frequency distribution, mean, median and mode. Standard deviation, standard error, t-, chi-square tests, correlation and regression. Environmental pollution: air pollution - pollutants – their sources, effects and control mechanisms. Water pollution - water pollutants – oxygen demanding wastes, pesticides, cultural eutrophication: sources, effects and control. Water treatment and wastewater treatment. Noise pollution: Sound, noise, noise measurement, noise classification, effects of noise pollution and control of noise pollution. Environmental Chemistry: chemistry of water, concept of DO, BOD, COD, Biochemical aspects of Arsenic, Cadmium, Lead, Mercury and Carbon Monoxide. Conventional, non-conventional energy sources, current issues and problems in environment.

BA 1261: UNIVERSAL HUMAN VALUES & PROFESSIONAL ETHICS, LAB – II, Credit: 1

Understanding Harmony in the Society, Harmony in Human, Human Relationship, Understanding Harmony in the Nature and Existence - Whole existence as Co-existence, Implications of the above Holistic Understanding of Harmony on Professional Ethics

CH 1163: ENGINEERING CHEMISTRY LAB, Credit: 1.5 (L-0, T-0,P-3)

12 labs are to be conducted on the basis of the syllabus of the corresponding theory paper

CS 1164: COMPUTER PROGRAMMING LAB 1.5 (L-0, T-0,P-3)

12 labs are to be conducted on the basis of the syllabus of the corresponding theory paper

Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,

SMIT, MAJITAR

REVISED SYLLABUS FOR B. TECH (CSE) COURSE CURRICULUM

APPLICABLE FOR 2019-20 BATCH FROM 2ND YEAR
APPLICABLE FOR 2020-21 BATCH FROM 1ST YEAR AND SUBSEQUENT BATCHES.

Total Credits:

$$22.5 + 22.5 + 23 + 22 + 20 + 12 = 122$$

PROGRAM EDUCATIONAL OBJECTIVES FOR B. TECH

COMPUTER SCIENCE AND ENGINEERING DEGREE COURSE

The AICTE-approved B. Tech. Computer Science and Engineering degree course at SMIT/SMU provides the education and training necessary to design, implement, test, and hone skill sets as per the requirement of the changing trends of industries. The curriculum elaborates on all aspects of computer systems from Logic Design, Computer Organization & Architecture, Data Structures, Operating System concepts, Networking, Higher-Level Language Skills, Object Oriented Programming Concepts, Database Management System, Software Engineering, including electives ranging from mobile computing to data analytics and their applications.

The Computer Science and Engineering graduates are prepared for employment in a wide spectrum of high-technology industries and also to inculcate them to become successful professionals. The curriculum lays solid foundation in Computer Science enabling the graduates to work with engineers from other disciplines. Graduates are sufficiently prepared to continue life-long learning and equip themselves for higher qualifications and research activities worldwide.

The Program Educational Objectives of Computer Engineering program are:

Engineering Knowledge: Our graduates will be capable of applying their engineering knowledge to succeed in whichever field they want to pursue keeping abreast of the ever-changing technology.

Entrepreneurship: Our graduates should be able to set up various entrepreneurship ventures which in turn facilitate employability.

Research Upliftment: Our graduates will apply the best practices of computation based on mathematics and science to address customized projects and ensure productivity in research.

Societal and Ethical Responsibility: Our graduates will showcase a sense of societal and ethical responsibility in their professional endeavors and should be able to make an informed choice for the furtherance of the society.

Cognitive Communication: Our graduates should be able to exhibit impromptu and impeccable communication skills with the potential of working as a team with cognitive empathy.

Department of Computer Science and Engineering

C210 **MA 1308** **(2L + 1 T hrs/week)**

DISCRETE MATHEMATICS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Course Objective: The objective of this course is familiarize on numerous counting techniques and abstract structures which appear frequently in many areas such as Algorithm analysis , data structures , database management system. Discrete mathematics plays a crucial role in enabling students of computer science to tackle these problems. Graph theory has tremendous application in Computer Networks, Switching. Group theory has enormous applications coding theory and related areas.

Course Outcome: On successful completion of this course, students should be able to:
1. Apply concepts of Group theory to the model real-world problems.
2. Analyze the use of graphs in Engineering applications.
3. Extend the concepts of Predicate Calculus in computer science like design of computing machines, artificial intelligence, definition of data structures for programming languages etc.
4. Evaluate problems related to mathematical logic and pigeonhole principle.
5. Solve problems using Permutation and Combination, Recursion and generating functions.

Pre-requisites: Differential & Integral calculus, Permutations and combinations.

UNIT I

Set theory, Group Theory and Graphs [18 Hrs]

Set theory: Principle of inclusion and exclusion, Relations, and functions, Techniques of Proofs, Pigeonhole Principle; Partial ordering, lattice and algebraic systems, principle of duality, basic properties of algebraic systems defined by lattices, distributive and complemented lattices.

Group Theory: Groups, subgroups, permutation group with simple examples. Cosets, normal subgroup, Burnside's theorem (statement only) and its simple applications, codes and group codes.

Graphs, Digraphs, Walk, Path, Cycles, Connectedness, Tree, Computer representation of relation, relation digraph, and graphs, transitive closer and Warshall's Algorithm.

UNIT II

Elementary configurations and Predicate calculus [18 Hrs]

Elementary configurations: - Permutations and Combinations, Generating functions, Partitions and Compositions, Lexicographical and Fike's orderings of permutations. Algorithms for Lexicographical, Reverse Lexicographical and Fike's ordering of permutation.

Predicate calculus: Connectives , Well formed formula (WFF), , Quantification, examples and properties of WFF into Causal form. Resolution and refutation, answer extraction and simple examples.

Text books :

I. Jean-Paul Tremblay and Manohar, R: Discrete Mathematical Structures with application to Computer Science, McGraw Hill.
2. C.L. Liu: Elements of discrete mathematics, McGraw Hill.
3. Narasingh Deo :Graph theory with applications to Computer Science , PHI.

References :

1. B. Kolman, R.C. Busby & S. Ross.: Discrete Mathematical Structures, Pearson
2. Principles of Artificial Intelligence; N. J. Nielson,
3. E. S. Page & L.B. Wilson: An Introduction to Computational Combinatorics, Cambridge University.

Department of Computer Science and Engineering

C201 **CS 1302** **(2L + 1 T hrs/week)**

DATA STRUCTURES

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: This course emphasizes on the organization of information, the implementation of linear data structures such as linked lists, stacks, queues, and non-linear data structures such as trees, and graphs. This course also explores recursion principles, the close relationship between data structures and algorithms and the analysis of algorithm complexity.

Pre-requisites: Programming concepts and 'C' language.

Course Outcome: On successful completion of this course, students should be able to:

1. Describe the working of data structures like array, stack, queue, linked list, tree and graph
2. Explain common applications for array, stack, queue, linked list, tree and graph
3. Solve a given problem using appropriate data structures and algorithm
4. Discuss about the working of the principal algorithms for sorting, searching, and hashing
5. Correlate the performance of a program with respect to the choice of data structure & algorithm

UNIT – I

Introduction [3 Hrs]
Definition, Algorithmic analysis: 'Oh' notation.

Contiguous data structures [4 Hrs]
Representation of multidimensional arrays, Highly structured sparse matrices using dimensioned arrays, **String representation and manipulation.**

Stacks [5 Hrs]
Definition, Operations on stacks, Implementation using array. Application of Stacks: Evaluation of arithmetic expressions. Recursion: Use of recursive techniques in enumeration problems and back tracking algorithms, Recursion removal using stacks.

Queues [5 Hrs]
Definition, Operations on queue, Implementation of queues, Circular queues, Applications.

Non-contiguous Data Structures [3 Hrs]
Linear linked list: Insertion, Traversal and deletion operations on singly linked list.Various types of linked list: Doubly linked list, Circular lists, Use of header node in circular lists, Generalized (recursive) list, Application of linear list, **Representation and manipulation of sets, Strings and graphs.**

UNIT- II

Trees [6 Hrs]
Definition of a tree and various terminologies used in tree, Binary tree, Recursive and non-recursive tree traversal algorithms, Representation of n-ary trees using binary trees, Application of trees, Expression trees. Search trees: Definition, Insertion, Deletion and reversal, Height balanced search trees (using AVL trees illustrative example) and weight balanced search trees.

Graphs [4 Hrs]
Terminology and representations: Introduction, Definition and terminology, Graph representations, Traversals, Connected components and spanning trees, Shortest path problem, Dijkstra's algorithm.

Sorting and searching [10 Hrs]

Sorting: Insertion, 2-way merge, Heap sort and quick sort, Comparison of different sorts, **Radix sort, Searching: Linear, Binary search, Comparison of different methods**. Hashing technique: Hash tables, Different hashing functions, Overflow handling, Methods for collision handling, Theoretical evaluation.

Text Books:

1. Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures", Galgotia.
2. Samanta, D., "Classic Data Structures", PHI.

Reference Books:

1. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson
2. E.M. Reingold and W.J. Hansen, "Data Structures", CBS.
3. A. S. Tanenbaum, Y. Langsam, M.J. Augenstein, "Data Structures using C", Pearson.
4. M. A. Weiss, "Data Structure and Algorithm Analysis in C", Pearson.

Department of Computer Science and Engineering

C203 **CS 1304** **(2L + 1 T hrs/week)**

DIGITAL CIRCUITS AND LOGIC DESIGN

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: It provides an in-depth coverage of digital circuits and logic design, starting from elementary 2-valued logic to expression minimization and circuit design techniques. This also includes the working principles of digital logic circuits and implementing them using logic gates, interfacing of logic families with TTL/ECL/MOS/CMOS etc.

Pre-requisites: Basic Electronics and Engineering Physics

Course Outcomes: On successful completion of this module, students should be able to
1. Relate and implement Boolean algebra in Digital Logic Circuits
2. Design combinational logic circuits
3. Design sequential logic circuits
4. Interpret various logic families
5. Explain the working of multivibrator circuits

UNIT – I

Logic gates and simplification of Boolean functions [6 Hrs]
Introduction to basic logic gates (AND, OR, NOT, NOR, NAND), The K-map method, SOP and POS simplifications, NAND and NOR implementations. Don't care conditions, Quine-McCluskey tabulation method (5-variable, decimal notation), **Determination and selection of prime implicants, MEV – techniques.**

Combinational logic [4 Hrs]
Design Procedure, Design of adders, Subtractor and code converters, Analysis procedure, Multi-level NAND and NOR circuits, Ex-OR and equivalence functions.

Combinational logic, MSI and LSI [4 Hrs]
Application of typical TTL IC components like binary parallel adder (74283), Carry look ahead adder, BCD adder (8283), Comparator (7485), Decoders (74138, 7442), Encoder (74148), Multiplexer (74157),parity generator.

Memory devices [6 Hrs]
Memory terminologies: **RAM, ROM, Word, Capacity, Address, Access Time, Cycle time, Magnetic core memory**, Semiconductor memory (S-RAM and D-RAM): Structures and operations, Read/Write cycles, Refreshing. Magnetic tape storage, Magnetic disk storage, Winchester disk and floppy disk, ROM and its architecture, **Types of ROM: PROM, EPROM, EEPROM, ROM applications, Linear selection and coincident selection.**

UNIT – II

Sequential logic [8 Hrs]
NAND/NOR gate latch, Clocked signals and clocked flip flops (S-R, J-K and D), J-K master-slave flip-flop,Edge triggering, Level Triggering , Ripple (Asynchronous) counters (mod 2N and mod<2N), IC asynchronous counters (7490, 7493), Asynchronous down counter, Cascading of counters, **Pre-settable counter (74193)**, Analysis of clocked sequential circuits, Design of clocked sequential circuits, State table and state transition diagram, sequential circuits design methodology.State reduction, State assignment, Flip-flop excitation tables, Design of synchronous counters,

Classification of sequential Circuits-Moore and Mealy Shift registers, **Universal shift register (74194)**, Ring counter, Johnson counter.

Integrated circuit logic families [6 Hrs]

IC Terminologies –I/O conditions, Fan in fan out, Propagation delay, Noise immunity, Circuits and characteristics of TTL, ECL, MOS (P-MOS, N-MOS), CMOS, **Interfacing of logic families to one another (TTL to CMOS, CMOS to TTL, TTL to ECL, and ECL to TTL).**

Multivibrators and timing circuits [6 Hrs]

Multivibrators- Bistable, Astable and monostable: Design of astable, Monostable, and bistablemultivibrators using transistors, Design of astable and monostablemultivibrators using logic gates, **Schmitt trigger circuit using IC 7413, Design of astablemultivibrator using IC 7413**.

Text Books:

1. Morris Mano, "Digital Logic and Computer Design", PHI
2. Tocci and Widmer, "Digital Systems", PHI
3. Herbert Taub and Donald Schilling, "Digital Integrated Electronics", McGraw Hill.

Reference Books:

1. Thomas C Bartee, "Computer Architecture and Logic Design", McGraw Hill
2. Louis Nashelsky, "Introduction to Digital Technology", Prentice Hall
3. Fletcher, "An Engineering Approach to Digital Design", PHI Raj Kamal, "Digital Systems Principles and Design", Pearson Education.

Department of Computer Science and Engineering

C204 **CS 1306** **(2L + 1 T hrs/week)**

COMPUTER ORGANIZATION AND ARCHITECTURE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: The main objectives of this course are to develop an understanding of the functional blocks of a computer and the inter-relation between them. It emphasizes on the design of the processing unit as well.

Pre-requisites: Digital Circuits & Logic Design, Computer System fundamentals.

Course Outcomes: On successful completion of this course, students should be able to:

1. Demonstrate competence in mapping relation between computer programming and computer organization and architecture.
2. Identify computer model requirements.
3. Analyze and evaluate the competence of a model.
4. Identify the evolution of Computer systems and the complexities in data processing.
5. Propose various microprocessor design alternatives to learn the concepts of parallel processing, pipelining and Interprocessor system performance

UNIT – I

Basic organization of the computer and Introduction to microprocessors [6 Hrs]
Basic organization of the computer and block level description of the functional units from program execution point of view, Fetch, Decode and Execute cycle. **Introduction to microprocessors, Evolution, a brief overview of more advanced processors (Pentium, Motorola and Zilog).**

Memory organization [6 Hrs]
A review of random and serial access memories, Basic concept of main memory: Static and dynamic memory, ROM, Error correction, Computer memory system overview, Memory hierarchy, Cache memory: Mapping functions, Replacement algorithms, Virtual memory, Logical to physical memory mapping, External memory: **Magnetic disk, RAID**.

Input/Output [4 Hrs]
External devices, I/O Modules, Programmed I/O, Interrupt driven I/O, DMA, I/O channels and processors.

The processing unit [4 Hrs]
Fundamental concepts: Fetching a word from memory, Storing a word in memory, Register transfers, Performing an arithmetic or logic operation Addressing modes, Instruction format: Three, Two, Oneand zero address instruction, **Control Unit:Hardwired control unit, Micro programmed control unit.**

UNIT- II

Arithmetic [7 Hrs]
Number representation, Fixed point addition and subtraction, Multiplication of fixed point numbers: Booth's multiplication, Integer division: Restoring and non-restoring.

Fundamentals of processor design [5 Hrs]
Instruction set processor design, Exploitation of instruction-level parallelism (ILP), Processor micro-architecture, Principles of processor performance, **Vector processing and array processing.**

Pipelined processor architecture [6 Hrs]
Fundamentals of pipelining, Flynn's classification of computers (SISD, SIMD, MISD, MIMD) Arithmetic pipeline design, Instruction pipeline design, Balancing pipeline stages, Stalls in pipeline, Methods for reduction of stalls in pipeline.

Reduced instruction set computers [2 Hrs]
Introduction, Reduced Instruction Set Architecture, **differences between RISC and CISC processors.**

Text Books:

1. V. C. Hamacher, Zaky, Vranesic, "Computer Organization", McGraw Hill
2. William Stallings, "Computer organization & Architecture – Designing for Performance", Pearson Education.
3. Gaonkar R.S.,"Microprocessor Architecture,Programming and Applications", 5th Ed., Penram International.

Reference Books:

1. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design - The Hardware/Software Interface", Morgan Kaufmann.
2. J. P. Hayes, "Computer Architecture and Organization", McGraw Hill.
3. Morris Mano, "Computer System Architecture", Pearson
4. P. Pal Chaudhuri, "Computer Organization and Design", PHI

Department of Computer Science and Engineering

C205 **CS 1307** **(2L+1T hrs/week)**

INTELLECTUAL PROPERTY RIGHTS & SOFTWARE ETHICS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective:
To introduce the role and importance of Intellectual property rights in the field of information technology
To understand the concepts of patents, trademarks and copyrights.
To introduce the procedure for obtaining patents, trademarks and copyright.

Pre-requisites: NIL.

Courses Outcomes: On successful completion of this course, students should be able to:

1. Identify regulations, legislation and standards for Intellectual Property Rights
2. Describe the impact of Intellectual Property Rights on engineering and industrial practices vis a vis social, environmental and economic context.
3. Apply principles of Intellectual Property Rights to sustainable design and development.
4. Analyze ethical lapses and recognize ethical dilemmas.
5. Distinguish professional issues which arise in the intellectual property law context

UNIT - I

Introduction to Intellectual Property Rights [8 Hrs]

Intellectual Property, Introduction to IPR, History of IPR, Overview & Importance, Role of IPR in Research & Development, Legislations Covering IPRs in India, Different forms of IPR – Patents, Copyright, Trademark, Industrial Designs, Layout Designs of Integrated Circuits, Geographical Indications, **Trade Secrets. Plant Varieties**, Some important examples of IPR.

Patents [6 Hrs]

Introduction, Patent – A form of property, The Patent Law in India, Patent Document, Protectable Subject matter: Patent and kind of inventions protected by a patent , Inventions which are not patentable under the act, Patent of addition, Term of patents of addition. Why protect inventions by patents? Searching for patents, **Drafting of a patent**. Filing of a patent. **Rights Conferred to Patentee.**

Copyright [6 Hrs]

Introduction: Meaning of copyright, Characteristics of copyrights, Indian Copyright Law. Main features of Copyright Act 1957, Amendments to Copyright Act, Requirements of copyrights, Copyright are protection in form and not in idea, Authorship and ownership of copyright. Rights conferred by copyright, Term of copyright, Related rights, Distinction between related rights and copyright.

UNIT II

Trademarks [4 Hrs]

Meaning of Trademark, the Functions of a Trademark, Essentials of a Trademark, Trademark Law in India, Domain name and how does it relates to trademarks? Registration of Trademark. Rights conferred by registration of Trademark.

IPR in the field of Information Technology [4 Hrs]

Introduction, Information Technology Act - 2000, Offences and corresponding penalties – Section 65 to 71, Section 66A and restriction of free speech, National Cyber Security Policy – 2013, Some notable cases.

Computer Ethics [4 Hrs]

Definition of Ethics, Computer and Information Ethics, The Ten Commandments of computer ethics, Hacking, **Ethical hacking**, Plagiarism.

Open Source Software [8 Hrs]

Introduction to Free and Open Source Software (FOSS), Open Source vs. Closed Source, Free Software, Free Software vs. Open Source software, **Copyright vs. Copyleft**, Licenses: GNU General Public License(GPL), MIT License, BSD License, Mozilla Public License, Apache License. Creative Commons, Public Domain, Forking Open Source projects; **Violation of copyrights and remedies. Using Open Source projects in industry. Open Source Government. Open Source Hardware. Open source media.**

Text Books:

1. Dr. B. L. Wadhera, Law Relating to Intellectual Property, Universal Law Publishing Co. Ltd.

2. P. Narayanan; Law of Copyright and Industrial Designs; Eastern Law House, Delhi

Reference Books:

1. Ajit Parulekar and Sarita D' Souza, Indian Patents Law – Legal & Business Implications; Macmillan India Ltd. 2006

2. National Cyber Security Policy, 2013.

3. Information Technology Act – 2000.

Department of Computer Science and Engineering

C211 **CS 1308** **(2L + 1 T hrs/week)**

OBJECT ORIENTED CONCEPTS & PROGRAMMING USING C++

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course introduces C++ as an Object Oriented Programming Language, building on existing knowledge of C and Java. The unit covers the C++language with a focus on its object-oriented features, and how these can be implemented as part of program designs and implementation. The students will also study and gain practical experience with the implementation issues related to object-oriented techniques, be able to build good quality software using object-oriented techniques, and understand the role of patterns in object-oriented design.

Pre-requisites: Computer Programming concepts

Course Outcomes: On successful completion of this course, students should be able to:

1. Describe the principles of Object Oriented Programming and C++ features that support OOP Paradigm.
2. Compare and differentiate between OOP and structured or procedural programming.
3. Explain programming constructs and features of C++ programming language.
4. Employ C++ constructs to write Object Oriented Programs to solve problems.
5. Adapt to bottom up approach of problem solving using classes and objects.

UNIT – I

Basic concepts of OOP [5 Hrs]
The Data types: Literal constant, Variables, Pointer types, String types, Constant qualifier, Reference types, The Boolean type, Enumeration types, Array types, Typedef names, Volatile qualifier, Class types, Expressions: Definition, **Operators: Arithmetic, Increment and decrement, Conditional, Size of operator,** New and delete, Comma,
The bitwise operator, Precedence, Type conversions, Statements: Selection, Iteration, Jump.

Procedural- based programming [4 Hrs]
Functions: Overview, Function prototype, Argument passing, Returning a value, Recursion, Inline functions, Linkage directives: Extern "C", Scope: Global objects and functions, Local objects, Dynamically allocated objects.

Object–based programming [5 Hrs]
Classes: Definition , Class object, Class member functions, Constructors and destructors, Friend function and classes, Static class members, Structures and unions, Bit field: A space saving member, Classscope, Nested classes, Structured programming and Object oriented programming paradigm.

Overloaded functions/ operators (polymorphism) [5 Hrs]
Overloaded declarations, The three steps of overload resolution, Argument type conversions, Overloading operators: Operators like =, ( ), ->, <<, >>, + and -- (Using both friend as well as member functions, Importance of this pointer).

UNIT- II

Generic programming using template function & template classes [6 Hrs]
Definition of generic programming, Function template, Template arguments and overloading function template.

Inheritance and io-stream library [8 Hrs]

Inheritance: Base class member access, Inheritance types, Protected mechanism, Virtual base classes. Virtual functions (Run-time polymorphism concept): Virtual functions, Pure virtual functions, Early vs. late binding. **The io-stream library: Streams, Stream classes, Managing unformatted and formatted I/O operations, Manipulators**.

Files/ exception handling [7 Hrs]

Classes for file stream operations, Accessing files, Sequential I/O operations, Random access, **Command- line arguments, Exception handling (throw, try, and catch).**

Text Books:

1. Herbert Schildt, "The Complete Reference C++", Tata McGraw Hill.
2. E. Balagurusamy, "Object Oriented Programming with C++", Tata McGraw Hill.

Reference Books:

1. Bjarne Stroustrup, "C++ Programming Language", Pearson.
2. Stanley B Lippman and Lajoie, "C++ Primer", Pearson.
3. Saurav Sahay, "OOP with C++", Oxford University Press.
4. B.L. Juneja and Anita Sethi, "Programming with C++", New Age International Publication.

Sharma

C206 **CS 1361** **(3 hrs/week)**

DATA STRUCTURES LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1302 Data Structures and associated prerequisites.

Course Outcomes: On successful completion of this course, students should be able to:

1. Write program related to application of data structures using programming constructs of a language like C programming language
2. Identify the suitable input and output for a specified problem statement
3. Conclude the working of various algorithms in terms of time and space complexity
4. Examine the errors encountered in the program using appropriate tools and fix them
5. Write well-indented and well-documented code

C207 **CS 1363** **(3 hrs/week)**

DIGITAL CIRCUITS AND LOGIC DESIGN LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1304 Digital Circuits and Logic Design, Basic Electronics and 10+2 level Physics.

Course Outcomes: On successful completion of this course, students should be able to:

1. Design and analyze basic combinational circuits.
2. Relate logic circuits to solve digital electronics problems.
3. Express flip-flops as memory elements and design digital electronics circuit with a memory.
4. Design small sequential circuits and implement them in software and hardware.
5. Design programmable counters and implement them in software and hardware.

C212 **CS 1365** **(3 hrs/week)**

OBJECT ORIENTED CONCEPTS & PROGRAMMING USING C++ LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1308 Object Oriented Concepts & Programming using C++ and associated prerequisites.

Course Outcomes: On successful completion of this course, students should be able to:

1. Tell the syntax and semantics of the C++ programming language.
2. Apply the concepts and principles of Object Oriented programming while writing programs using C++.
3. Extend the concepts of encapsulation, polymorphism, inheritance using class, objects, function/operator overloading, function overriding, etc.
4. Write programs with Inheritance, virtual functions that supports code reusability, dynamic binding and run time polymorphism.
5. Adapt to bottom up approach of problem solving using classes and objects.

Department of Computer Science and Engineering

C230 **MA 1408** **(2L+1T hrs/week)**

NUMERICAL METHODS, COMPLEX AND FOURIER ANALYSIS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: The objective of this course is to equip the students of Computer Science with computational aspects of many physical problems. Most of these problems do not possess exact or analytical solution. These problems are then solved by means of efficient numerical techniques. The emphasis will be more on algorithmic approach on these methods. Fourier transforms play an important role in many areas of computer science such as Image processing, pattern recognition. The course has been designed to empower the students to apply the concepts of Fourier analysis and transforms to different areas of their core courses. Complex analysis plays a critical part in most engineering branches, this module has been framed in such a way that it will have wide applications.

Course Outcome: On successful completion of this course, students should be able to:
1. Create ability to handle complex integrations appearing in different engineering areas.
2. Apply the concepts of interpolation to find best Curve fitting for given data and also evaluate Integration and differentiation numerically.
3. Solve differential equations numerically.
4. Evaluate solution of algebraic and Transcendental equations and system of linear equations using iterative methods.
5. Associate between the concepts of Fourier analysis and applications in the field of Signal processing, Image processing etc.

Pre-requisites: Differential & Integral calculus, differential equations, Linear algebra, Laplace transformations.

UNIT I

Numerical Methods [18 Hrs]
Introduction to Error, Interpolation and application: finite difference, central and divided differences, Newton-Gregory, Lagrange's, Dividend interpolation formulae. Numerical differentiation and integration: Trapezoidal rule, Simpson's one third rule. Solution of systems of linear equation: Jacobi, Gauss-Seidal, Solution of tridiagonal systems, computation of largest eigen value by power method.
Numerical solution of algebraic and transcendental equations using method of Regula Falsi and Newton Raphson's method.
Numerical solution of initial value problems in ordinary differential equations by Taylor series method, Picard's method, Euler's method, Modified Euler's method, Runge Kutta fourth order method, Milne's Predictor and Corrector method.

UNIT II

Complex Analysis [12 Hrs]
Complex functions, analyticity -Cauchy Riemann equations. Cauchy's integral theorem and Cauchy's integral formula , derivatives of analytic functions. Taylor, Maclaurin and Laurent's series, residue theorem, evaluation of standard real integrals using contour integrals.

Fourier Transforms [6 Hrs]
Fourier Series, Fourier Integrals, Fourier Transformations; sine and cosine transforms.

Text books:
1. Conte,S & deBoor: Elementary Numerical Analysis, An algorithmic approach, McGraw Hill.
2. S.S. Sastry : Introductory methods of numerical analysis, Prentice Hall India.

3. V Rajaraman: Computer Oriented Numerical Methods, PHI.
4. R. V Churchil & Brown: Complex variables and its applications. TMH.
5. Erwin Kreyszig:, Advanced Engineering Mathematics, Wiley Eastern.

References :

1. Jain & Iyengar Jain: Numerical Methods, New Age publications
2. Kasana : Complex variables ; Theory and Applications
3. B. S. Grewal: Higher Engineering Mathematics.

Department of Computer Science and Engineering

C220 **CS 1403** **(2L+1T hrs/week)**

DATABASE MANAGEMENT SYSTEMS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: This course provides the basic information about relational Database Management System and their development. The major objectives of the course is to provide an introduction of DBMS and their use, be familiar with the basic DBMS architecture, components, and interfaces, have experience using at least one modern Database Management System, understand and use database models in database and application design

Pre-requisites: Programming Concepts

Course Outcomes: On successful completion of this course, students should be able to:

1. Describe fundamental elements of a relational database management system.
2. Design entity-relationship diagrams to represent simple database application scenarios
3. Explain the basic concepts of relational data model, Entity-relationship model, Relational database design, relational algebra and database language SQL
4. Apply and relate the concept of transaction, concurrency control and recovery in database
5. Analyze various Normalization techniques and improve the database design by normalization

UNIT- I

Introduction [4 Hrs]
DBMS: Characteristics, Advantages, Architecture. Database concept and architecture, Data models, Instances and schema, Database languages, Database manager, Database administrator, Database users, Concept of centralized database management system and distributed database system.

Data modelling [6 Hrs]
Entity sets, attributes types and keys, Entity Relationship (ER) diagram, Type role and structural constraints, Enhanced entity-relationship (EER), Object modelling, Specialization and generalization, Modelling of union types, **Data models: Definition, Purpose and Types, Hierarchical models, Network model, Relational model,** Relational-algebra operations.

Database design [7 Hrs]
Database design process, Relational database design, Relation schema, Functional dependencies, Membership and minimal covers, Normal forms, Multivalued dependencies, Join dependencies, **Converting EER diagrams to relations, Effect of de-normalization on database performance**.

Data Storage and Indexes [3 Hrs]
File organizations, primary, secondary index structures, various index structures - hash-based, dynamic hashing techniques, multi-level indexes, B+ trees.

UNIT-II

Database query languages [4 Hrs]
Query-by-example (QBE), Introduction to SQL, Use of some special data types, Overview of SQL 92, Basic queries in SQL, Advanced queries in SQL, Functions in SQL, Basic data retrieval, Aggregation, Categorization, Updates in SQL, Embedded SQL and 4GLs, Procedural extension to SQL: PL/SQL.

Transaction processing [5 Hrs]
Desirable properties of transactions, Implementation of atomicity and durability, Reconsistent model, Read only and write only model, Concurrent executions, Schedules and recoverability, Serializability of schedules, Concurrency control, Precedence graph.

Concurrency control and backup & recovery mechanisms [5 Hrs]
Overview of concurrency control, Locking techniques, Lock based protocols, Time stamp based protocols, Commit protocols, Optimistic technique, Granularity of data items, Time stamp ordering multi version concurrency control, Deadlock handling, Recovery mechanisms, Database recovery techniques based on immediate and deferred update, **Concepts of database security mechanisms, Case study of Distributed Database Systems.**

Graph Database [3 Hrs]
Overview of graph database, Structure and advantages of graph database, **high level view of graph space, Property graph model.**

NoSQL [3 Hrs]
An overview of NoSQL, Characteristics of NoSQL, Advantages and challenges of NoSQL, NoSQL storage types, Case study of MongoDB.

Text Books:
1. Elmasri and Navathe: "Fundamentals of Database Systems", Addison Wesley.
2. Silberschatz, Korth, Sudarshan, "Database System Concepts", McGraw-Hill.

Reference Books:
1. Thomas Connolly, Carolyn Begg, "Database Systems – A Practical Approach to Design, Implementation and Management", Pearson Education.
2. Jefrey D. Ullman, Jenifer Widom, "A First Course in Database Systems", Pearson Education.
3. Bipin C Desai, "An Introduction to Database Systems", Galgotia.
4. Atul Kahate, "Introduction to Database Management Systems", Pearson.
5. Ian Robinson, Jim Webber, Emil Eifrem,"Graph Databases",O'Reilly Media.
6. Gaurav Vaish, "Getting started with NoSQL", Packt.

Department of Computer Science and Engineering

C221

CS 1405 (2L+1T hrs/week)
DESIGN AND ANALYSIS OF ALGORITHMS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: This course builds upon preliminary knowledge delivered in Data Structures. The main objectives of the course are to provide thorough knowledge and understanding of different algorithm analysis techniques, design strategies and their applications. Special purpose machines, some critical problems and innovative techniques are used in solving them.

Pre-requisites: Data Structures and Programming concepts

Course Outcomes: On successful completion of this course, students should be able to:

1. Define asymptotic notations and solve problems related to it
2. Calculate time and space complexities for recursive/non-recursive algorithm based on following algorithm design techniques - divide and conquer, greedy, dynamic programming and branch and bound.
3. Select appropriate algorithm design technique to solve a given problem.
4. Explain the working of existing algorithm / algorithm design techniques
5. Discuss and describe the classes P, NP, and NP-Complete

UNIT – I

Algorithms [4 Hrs]
Definition, Aim of the subject, Designing algorithms and Analyzing algorithms: An introduction, Performance of a program: Space and Time complexity.

Mathematical preliminaries [4 Hrs]
Asymptotic notations and common functions, Example: Insertion sort

Recurrences and divide and conquer [7 Hrs]
The basics of divide & conquer method, Merge sort , Quick sort, Solving recurrences: Substitution method, Recursion tree method, Master method: **Proof of master method**, **Finding maximum and minimum**, Strassen's matrix multiplication, **Binary search**.

Greedy method [7 Hrs]
Basics of greedy method, Applications- 0/1 Knapsack Problem – Topological sorting – **Bipartite Cover**, Heapsort, Huffman codes, **Activity selection**, Minimum spanning tree-Kruskal's algorithm, Prim's algorithm, Single source shortest path: Dijkstra's algorithm.

UNIT – II

Dynamic programming [7 Hrs]
Basics of dynamic programming, Applications- Matrix chain multiplication, Longest common subsequence, Traveling salesperson problem, all pair shortest path-Floyd and Wars hall's algorithm, **Non Crossing Subsets of Nets**.

Back Tracking [5 Hrs]
Backtracking Method, Applications-**Container Loading**, 0/1 Knapsack Problem, **Max Clique**, Travelling Salesperson, Board Permutations

Branch And Bound [5 Hrs]

Branch and Bound Method, Applications-**Container Loading**, 0/1 Knapsack Problem, **Max Clique** ,Travelling Salesperson , **Board Permutations**

NP completeness [2 Hrs]

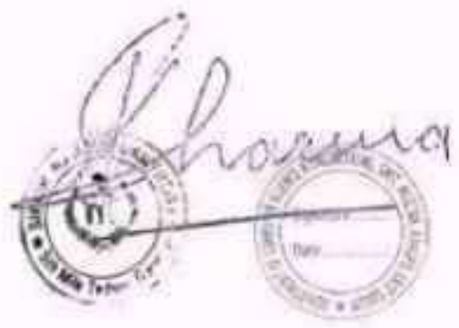
Basic Concepts, P NP, NP Complete, NP Hard problems, Travelling Salesman Problem.

Text Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Introduction to Algorithms", PHI.
2. Sartaj Sahni, "Data Structures, Algorithms amd Applications in C++ ", University Press

Reference Books:

1. A. Levitin, "Introduction to the Design and Analysis of Algorithms", Pearson Education
2. S. Basse, A. Van Gelder, "Computer Algorithms-Introduction to Design and Analysis", Pearson
3. Alfred V. Aho, John E. Hop croft, Jeffrey D. Ullman, "Data Structures and Algorithms", Addison Wesley.
4. M. A. Weiss, "Data Structure and Algorithm Analysis in C", Pearson Education.



Department of Computer Science and Engineering

C222 **CS 1406** **(2L+1T hrs/week)**

ADVANCED COMPUTER ORGANIZATION AND ARCHITECTURE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objectives: This course focuses on advanced computer organization and architectures such as pipelined and parallel systems. It also emphasizes on the concept of a complete system consisting of asynchronous interactions between concurrently executing hardware components and device driver software in order to illustrate the behavior of a computer system as a whole.

Pre-requisites: Computer Organization and Architecture.

Course Outcomes: On successful completion of this course, students should be able to:

1. Review Computer System Architecture.
2. Evaluate complexities in data representation and processing.
3. Apply concepts of parallel processing and multiprocessor architectures in reviewing processors.
4. Solve problems related to multiprocessing, distributed processing and non von Neumann architectures.
5. Illustrate the concept of data flow computers, Reduction computer architecture and systolic architecture

UNIT – I

Introduction to parallel processing [5 Hrs]
Criteria for judging the architecture, **Architectural classification schemes**, Trends towards parallel processing, Parallelism in uniprocessor systems, Parallel computer structure, Performance evaluation of Processors – Amdahl's law, **Applications of parallel processing.**

Principles of pipelining [5 Hrs]
Principles of linear and non-linear pipelining, **Classification of pipeline processors**, General pipelines and reservation tables, Interleaved memory organization.

Instruction-level parallelism [5 Hrs]
Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP; Superscalar, super-pipelined and VLIW processor architectures; **Vector and symbolic processors**; **Case studies of contemporary microprocessors**

Structures and algorithms for array processors [5 Hrs]
SIMD array processors: SIMD computer organization, **Masking and data routing mechanisms**, SIMD interconnection networks: static v/s dynamic, Mesh connected ILLIAC network, Barrel shifter network, Shuffle-exchange and Omega network., Sequential matrix multiplication, Algorithm for processorarray-2D mesh SIMD model, Hypercube SIMD model, Shuffle exchange mode.

UNIT – II

Multiprocessor architecture [6 Hrs]
Functional structures, UMA and NUMA multiprocessors, Interconnection Networks: **Time shared or common buses,** Bus arbitration algorithm, Cross bar switch and multiport memories, Comparison of multiprocessor interconnection structure, multistage networks for multiprocessors.

Elementary parallel algorithms [6 Hrs]
Developing algorithms for processor arrays/MIM computers, Process Communication and synchronization on MIMD, **Deadlock**, Task scheduling on MIMD.

Multiprocessor Architecture [5 Hrs] Taxonomy of parallel architectures; Centralized shared-memory architecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, **Cluster computers, Case Study: Intel Montecito and Sun Niagara**.

Non von Neumann Architectures [3 Hrs] Data flow Computers, **Reduction computer architectures, Systolic Architectures**.

Text Books:

1. Kaihwang and Faye A. Briggs, "Computer Architecture and Parallel Processing", McGraw Hill.
2. Michael J. Quinn, "Parallel Computing: Theory and Practice", McGraw Hill.
3. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.

Reference Books:

1. Kaihwang, "Advanced Computer Architecture – Parallelism, Scalability, Programmability", Tata McGraw Hill.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson.
3. Michael J.Quinn, "Parallel Computing Theory and Practice", McGraw Hill.
4. Rajiv Chopra, "Advanced Computer Architecture", S. Chand Group.

Department of Computer Science and Engineering

CS 14** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE I

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question
Objectives and Pre-requisites are given under Elective details
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 14** **(2L+1T hrs/week)**

OPEN ELECTIVE I

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question
Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

C225 **CS 1462** **(3 hrs/week)**

DATABASE MANAGEMENT SYSTEM LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.
Pre-requisites: Corresponding theory paper CS 1403 Database Management Systems and associated prerequisites.
Course Outcomes: On successful completion of this course, the student should be able to:

1. Select appropriate SQL/MongoDB commands and functions for a given query on the database.
2. Infer constraints and relationships between tables from conceptual/logical level schema and convert them into relationship and integrity constraints at the physical level schema.
3. Write Oracle PL/SQL Programs for data processing.
4. Design nested queries for efficient data processing on the database.
5. Test an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.

Department of Computer Science and Engineering

C226 **CS 1464** **(3 hrs/week)**

ADVANCED PROGRAMMING LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Basic programming languages (C and C++).

Course Outcomes: On successful completion of this course, the student should be able to:

1. Discuss the concept of objects and class.
2. Evaluate requirements for given problem and decide the functionalities of programs accordingly.
3. Illustrate a diverse set of problem solutions using techniques of Interface, Packages, File Handling, Multi-threading, etc
4. Apply the concept java network program to establish connection between client and server.
5. Design modern website to fulfill user's requirement.

C227 **CS 1465** **(3 hrs/week)**

ALGORITHM LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1405 Design and Analysis of Algorithms and associated prerequisites.

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

1. Calculate time and space complexities for algorithms using mathematical models
2. Calculate time and space complexities for algorithms by performing an empirical measurement
3. Select appropriate algorithm design technique to solve a given problem
4. Write a computer program to implement any algorithm using a programming language
5. Choose the most optimal algorithm by comparing time and space complexities

Signature
Date

Department of Computer Science and Engineering

C312 **MA 1502** **(2L + 1 T hrs/week)**

PROBABILITY, STATISTICS AND STOCHASTIC PROCESSES

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Course Objectives: The objective of this course is to enhance the capability of students to analyze the problems related to random phenomena. Concepts on probability theory will be of immense help to the students in analyze random experiments. Statistical Analysis plays a big role in areas like data mining and information retrieval. Stochastic models have tremendous applications in queuing theory. Students will find adequate tools in these modules which will be effective enough to solve their problems.

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

1. Explain concept of probability, random variables.
2. Differentiate the ideas between discrete and continuous random variables.
3. Discuss fundamentals of probability and statistical theories.
4. Apply the knowledge of probability and statistics to analyze different real-world situations
5. Develop the models of many time dependent processes such as signals in communications , time series analysis.

Pre-requisite: Differential and Integral Calculus, Matrix Algebra, Permutation and Combination.

UNIT I

Probability Theory [18 Hrs]

Introduction, Classical definition of probability, Axiomatic definition of probability, Conditional probability, Baye's theorem. Random variable, Discrete random variable, Bernoulli trials, Poisson trials, Discrete distribution: Binomial, Poisson, Continuous distribution: Uniform, Normal, exponential. Expectation: Mean, Variance, Chebyshev's inequality, Central limit theorem. Two dimensional random variables: discrete and continuous, marginal distributions, Covariance, Correlation coefficient, conditional distributions, conditional expectations. Reliability & MTTF.

UNIT II

Statistics [6 Hrs]

Random sample, Sampling distribution, Statistic, Least square curve fitting Parameter estimation: Unbiased estimate, Consistent estimate, Maximum likelihood estimate, interval estimate. Testing of Hypothesis for mean with known variance for normal population.

Stochastic process & Queuing theory [12 Hrs]

Introduction to Stochastic Process, Poisson Process, Discrete parameter Markov Chains, Concept of a queues: Basic idea of continuous parameter Markov chain, Birth and death processes, M/M/1/∞, M/M/1/N queuing systems. (Approach for entire stochastic process will be more problem oriented).

Textbook:

1. K.S Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, 2008, Wiley.
2. P.L. Meyer : Introductory Probability theory and statistical Applications, Second Ed. Oxford & IBM Publishers.
3. Miller & Freund: Probability & Statistics for Engineers, Eight Ed. Pearson Ltd.

Reference Books:

1. Introduction to Probability Theory and its Applications, William Feller, 2008, Wiley.
2. Introduction to Probability with Statistical Applications, Geza Schay, 2007, Brikhaus.

Department of Computer Science and Engineering

C301 **CS 1502** **(2L + 1 T hrs/week)**

OPERATING SYSTEMS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: The principles and concepts that govern the design of modern computer operating systems are studied. Managing computing resources such as the memory, the processor and the Input/output devices are covered. Algorithms for CPU scheduling, memory and general resource allocation; process coordination and management; deadlocks and memory management techniques; case studies of Linux operating systems are also covered.

Pre-requisites: Computer Organization and Programming Language concepts.

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

1. Describe the different operating system architectures and structures.
2. Select the appropriate scheduling algorithms or techniques for efficient utilization of computer resource like CPU, Memory, etc.
3. Interpret theory of data inconsistency problem and provide a procedure for synchronization problem.
4. Manage various technical issues related to operating systems' services using principles of computer science and engineering.
5. Compose methods for analyzing the performance of various identified algorithms or techniques in operating systems

UNIT – I

Introduction [4 Hrs]
Basics of Computer Organization, Interrupt, Bus, ISA, CPU Operation. What operating systems do? Operating system structure, Operating system operations, Special-purpose systems, Operating system services, User-operating system interface, System calls, Types of system calls, **Operating system design and implementation. Case study: Linux Design Principles.**

Process management [8 Hrs]
Process: Concept, Multithreaded programming, Multithreaded models, Thread libraries, Threading issues, Process scheduling criteria and algorithms, Thread scheduling, Operating-system examples. **Case study: Linux Process and I/O Scheduler.**

Process synchronization [8 Hrs]
Critical section problem, Peterson's solution, Synchronization hardware, Semaphores, Classic problems, **Monitors, Examples.**

UNIT-II

Deadlock [5 Hrs]
System model, Characterization, Methods of handling deadlocks, Prevention, Avoidance, Detection and recovery.

Memory management [5 Hrs]
Fixed and variable partition, Swapping, Paging and segmentation, Structure of page table, Combined systems, Virtual memory: Overlays, Demand paging, Copy-on-write, Page replacement, Allocation of frames, Thrashing, Allocating kernel memory. Case study: Linux Memory Management

File systems [5 Hrs]

File system concept, Access methods, Directory structure, File-system mounting, File sharing, Protection, Allocation methods.

I/O systems [5 Hrs]

Overview, Kernel I/O subsystem, Kernel Data Structure, Transforming I/O requests to Hardware Operations, **Case Study: Linux I/O Systems.**

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Wiley & Sons.Inc..
2. D M Dhamdhere, "Systems Programming & Operating Systems", Tata McGraw-Hill.

Reference Books:

1. Andrew S. Tanenbaum, "Modern Operating systems", PHI.
2. Mukesh Singhal, Niranjan G.Shivaratri, "Advanced Concepts in Operating Systems", Tata McGraw- Hill.
3. P. Balakrishna Prasad, "Operating Systems", Scitech Publication.
4. William Stallings, "Operating Systems-Internals and Design Principles", Pearson Education.

Department of Computer Science and Engineering

C305 **CS 1508** **(2L + 1 T hrs/week)**

COMPUTER NETWORKS-I

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course aims to highlight the functional evolution and role of data communications. Considerations in data communications, Applications in general, Design issues, System components and their interrelationships, Networks. Synchronous and asynchronous communications, Implementation of data communication in Physical layer and Data Link layer.

Pre-requisites: Communication Techniques.

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

1. Understand and identify the engineering fundamentals concerning Data Communication and Computer Network.
2. Identify the complex engineering problem involved in Data communication and its architecture for a successful network based communication.
3. Ability to Practice and formulate a solution for an engineering problem concerning any layers in Data Communication model
4. Demonstrate an ability to formulate and interpret a model based on the fundamentals of Computer Networks
5. Compare and understand the IOS reference model and TCP-IP reference model for data communication

UNIT- I

Overview [3 Hrs]
Introduction to data communications, Data representation and data flow, Concepts of communication in computer networks, Layered architecture, OSI, TCP/IP, **ATM network models**, Addressing of network devices.

Physical layer and Transmission Media [6 Hrs]
Data and signal fundamentals, Analog and digital signals, Transmission impairments, Data rate limits, Performance, Guided media: Characteristics and performance parameters of various twisted pair, Coaxial, and fibre optic cables. Unguided media: Radio waves, Microwaves and infra-red.

Data transmission [6 Hrs]
Digital encoding techniques, Scrambling techniques, Pulse Code Modulation (PCM), Modulation, Transmission modes (parallel, serial).

Multiplexing and Switching methods [5 Hrs]
FDM, WDM, CDM, Time division multiplexing (synchronous and statistical), Spread spectrum (FHSS and DSSS), Circuit-switched, Datagram and virtual circuit networks, **Message switching**.

UNIT- II

Data link layer [8 Hrs]
Data link layer design issue, Error detection and correction: Parity bit, Modulo-2 arithmetic, Polynomial, FEC-Hamming code, Internet checksum, Elementary data link protocol: Stop-and-wait ARQ, Sliding

window, Go-back-n, Selective repeat. Random Access: CSMA, CSMA/CA, CSMA/CD, **Controlled Access: Reservation, Polling, Channelization: FDMA, TDMA, CDMA.**

Network layer [7 Hrs]
Design issues of network layer protocols, Network layer protocols of TCP/IP model: Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6), Addressing mechanism of IPv4 and IPv6.

Case study of popular LANs [5 Hrs]
Ethernet LAN: IEEE specification of physical media, Wireless LAN: IEEE 802.11 specification, **WLAN architecture.**

Text Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw Hill.
2. William Stallings, "Data and Computer Communications", PHI.

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", PHI.
2. A S Godbole, "Data Communication and Networking", Tata McGraw Hill.
3. William C Y Lee, "Mobile Communication Engineering", Tata McGraw Hill.
4. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Wiley.

Department of Computer Science and Engineering

C313

CS 1509
SOFTWARE ENGINEERING

(2L + 1 T hrs/week)

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: This course presents a comprehensive study of software quality assurance, including software quality control management, processes, systems, methods, standards, certification, and reliability measurement.

Pre-requisites: Procedure oriented and object oriented programming paradigm.

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

1. Demonstrate competence in using engineering fundamentals to visualize solutions using knowledge of software engineering skills.
2. Extend an ability to formulate a solution plan and methodology for an engineering problem using software engineering.
3. Apply an ability to formulate and interpret a model for project management
4. Explain an ability to define complex problem, find and analyze requirements
5. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.

UNIT – I

Introduction [2 Hrs]
The software engineering discipline-evaluation and impact, Programs vs. software products, Emergence of software engineering, Notable changes in software development practice, System engineering, handling complexity through Abstraction and Decomposition.

Software Life Cycle [6 Hrs]
Life Cycle Models: Classical waterfall model, Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, RAD model, Agile models, Introduction to Agile software development, Extreme Programming and Scrum, DevOps, Domain Analysis, **Comparison of different life cycle models.**

Software Project Management [6 Hrs]
Responsibilities of project manager, Project planning, Metrics for project size estimation techniques, Empirical estimation techniques, COCOMO, Halstead's software science, Staffing level estimation, Scheduling, **Organization and team structure**, Staffing, Risk management, Software configuration management.

Requirements Analysis and Specification [2 Hrs]
Requirements gathering and analysis, Software requirement specification (SRS), Traceability, **Characteristics of a Good SRS Document, IEEE 830 guidelines**, Overview of formal system development techniques.

Software Design [4 Hrs]
Good Software Design, Cohesion and coupling, Control Hierarchy: Layering, Control Abstraction, Depth and width, Fan-out, Fan-in, Software design approaches, object oriented vs. function oriented design.

UNIT – II

Function-Oriented and Object Oriented Software Design [5 Hrs]
Overview of SA/SD methodology, structured analysis, Data flow diagram, Extending DFD technique to real life systems, structured design, detailed design, Design review. Unified Modeling Language (UML), UML Diagrams: Static and Dynamic.

User interface design [3 Hrs]
Characteristics of a Good User Interface, User Guidance and Online Help, Mode-based Vs Mode-less Interface, Types of user interfaces, Component-based GUI development, User interface design methodology: GUI design methodology, Task and object modeling, Selecting a metaphor, Interaction design and rough layout, **User interface inspection**.

Coding and Testing [5 Hrs]
Coding, Code review, Testing-Basic Concept of testing, Testing Strategies, Testing in the large vs. testing in the small, Unit testing, Black-box testing, Integration testing, System testing, Some general issues associated with testing, Test driven development, testing tools, Special Value Testing, Combinatorial Testing, Decision Table Testing, Cause effect graphing, Pairwise Testing, White box Testing, Condition Testing, MC/DC Coverage, MC/DC Testing, Path Testing, Dataflow and Mutation Testing, Debugging, Program analysis tools,

Software Reliability And Quality Management [3 Hrs]
Software reliability, Statistical testing, Software quality and management, ISO 9000, SEI capability maturity model, Personal software process (PSP), Six sigma, Software quality metrics

Computer Aided Software Engineering [2 Hrs]
Case and its scope, Case environment, Case support in software life cycle, **Other characteristics of case tools, Towards second generation case tool, Architecture of a case environment.**

Software Maintenance and Reuse [2 Hrs]
Characteristics of software maintenance, Software reverse engineering, Software maintenance processes model, Estimation maintenance cost. Basics issues in any reuse program, **Reuse approach, Reuse at organization level.**

Text Book:
1. Rajib Mall, "Fundamentals of Software Engineering", PHI.
2. Richard Fairley, "Software Engineering Concepts", Tata McGraw Hill.

Reference Books:
1. Jalote Pankaj, "An integrated approach to Software Engineering", Narosa.
2. Pressman R, "Software Engineering- Practioner Approach", McGraw Hill.
3. Somerville, "Software Engineering", Pearson
4. Budgen, "Software Design", Pearson

Department of Computer Science and Engineering

CS 15** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE-II

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 15** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE-III

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

C307 **CS 1561** **(3 hrs/week)**

OPERATING SYSTEMS LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1502 Operating Systems and the associated prerequisites

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

1. Illustrate the concept of process and thread creation for executing user's task.
2. Apply the theory for implementing various process scheduling algorithms.
3. Produce a optimal solution for data inconsistency problem by synchronizing processes and threads.
4. Examine the various memory management strategies for efficient resource utilization and implement it.
5. Analyse and debug various technical issues related to operating systems services and use different types of Modern OS

Department of Computer Science and Engineering

C331 **CS 1604** **(3L + 1 T hrs/week)**

FORMAL LANGUAGES AND AUTOMATA THEORY

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

OBJECTIVES: This course builds upon preliminary knowledge delivered in discrete structure for computer science and computer programming concepts. The main objectives of the course are to provide learners with a detailed understanding of the mathematical models of the machines and their evolution through requirement generation and advancement in languages. Thorough the concepts and operations in Formal Language and Automata Theory, their use in Compiler Design and their application in Natural Language Processing.

PRE-REQUISITES: Knowledge in discrete mathematics and in programming.

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

1. Explain different concepts in automata theory and formal languages.
2. Produce various grammars and their acceptors.
3. Analyze the various language acceptors.
4. Acquire a fundamental understanding of computational models related to decidability and recursive enumerability.
5. Illustrate various proofs using mathematical principles.

UNIT I

Introduction [2 Hrs]
Mathematical preliminaries: Sets, Logic, Functions, Relations, Languages. Definitions: Language, Grammar, Automata, Relation between language, Grammar and automata, Importance of automata theory.

Finite Automata [7Hrs]
Informal introduction: Drawing examples from everyday life to bring out the essence of finite automata, Finiteness and its importance in automata theory. Deterministic finite automata: Definition, Processing strings, Transition functions, Language of a DFA; Nondeterministic finite automata: Non-determinism, Definition, Extended transition functions, Language of a NFA, Equivalence of DFA and NFA, Kleene's theorem, Epsilon transitions, Applications of Finite automata in text search.

Regular Expressions and Regular Languages [6 Hrs]
Memory required to recognize a language, Regular expressions, Regular expression to finite automata, Finite automata to regular expression, Algebraic laws for regular expressions, applications of regular expressions, Criterion for regularity, Regular languages.

Properties Of Regular Languages [3 Hrs]
Pigeonhole principle, Pumping lemma for regular languages, Closure properties, **Testing membership of regular languages, Equivalence of automata.**

UNIT II

Context Free Grammars and Languages [6 Hrs]
Definition, Leftmost and rightmost grammars, Parse trees, Ambiguity: Ambiguous grammar, removing ambiguity. Normal forms, Applications of context free grammars: Parsers

Pushdown Automata and Context Free Languages [4 Hrs]

Definition of pushdown automata, Representing pushdown automata, Acceptance by pushdown automata: By final state, By empty stack, Deterministic pushdown automata, Equivalence of pushdown automata and context free grammars, **Pumping lemma for context free languages. Closure properties of context free languages.** Testing membership of context free, Decision problems for context free languages.

Turing Machines [5 Hrs]

Definition, Language of a Turing Machine, Programming Turing Machines, The Church-Turing Thesis, A simple programming language, Extensions of the Basic Turing Machine.

Recursively Enumerable Languages [2 Hrs]

Definition, Enumeration, Chomsky hierarchy.

Undecidability [3 Hrs]

The halting problem, the post correspondence problem,**Time and space complexity of Turing machines. Complexity classes**.

Language Learning [2 Hrs]

Learning framework. Inductive inference. Grammar induction

1. **Text Books:**

(T1): John. E. Hopcroft, Rajeev Motwani, Jeffry.Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education.

(T2): Peter Linz, An Introduction to Formal Languages and Automata, Narosa

2. **Reference Books:**

(R1): James. L. Hein, Discrete Structures, Logic and Computability, Narosa

(R2): Partha Niyogi, The Computational Nature of Language Learning and Evolution, PHI.

(R3): C.K. Nagpal, Formal Languages and Automata Theory, Oxford University Press, 2011.

(R4): John Martin , Introduction to Languages and the Theory of Computation, Tata McGraw Hill.

Department of Computer Science and Engineering

C333

CS 1606 **(3L + 1 T hrs/week)**

COMPUTER NETWORKS-II

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course emphasizes on several computer network concepts, applications and protocols in OSI as well as TCP/IP layered architecture. It also covers the various protocols of various layers, their operations and applications. Further it discusses the concept of network security, challenges and their counter measures.

Pre-requisites: Communication Technique and Data Communication.

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

1. Understand and discover the engineering fundamentals involved in Computer Network and other related frame work
2. Identify the complex engineering problem relating computer network relating to host Identification data delivery and routing.
3. Ability to formulate a solution plan and methodology for an engineering problem concerning Computer Networking like Sub network, super network, and DNS etc
4. Ability to formulate and interpret a model based on the Computer Networks and its related framework.
5. Understand and explain basic responsibilities/concept of protocols in protocol present stack, DNS, DHCP, security, etc.

UNIT- I

Network layer [6 Hrs]

IPv6 packets and addressing, Internet Control Message Protocol (ICMP), Internet Group Message Protocol (IGMP), Mapping Physical to Logical Address: Reverse Address Resolution Protocol (RARP) Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP).

Routing protocols for wired network [6 Hrs]

Unicast routing protocols: Shortest Path, Flooding, Distance Vector routing (DVR), Link state routing, Multi cast routing protocols. Interior gateway protocol: Open Shortest Path First (OSPF), Exterior gateway protocol: Border Gateway Protocol (BGP).

Transport layer [8 Hrs]

Functions of transport layer protocols: Congestion control, Reliable service, Introduction to Transmission Control Protocol (TCP) as Transport Layer Protocol, Header description, Congestion control mechanism of TCP, Transport Protocols User Datagram Protocol (UDP), Use of UDP, Header description, Stream Control Transmission Protocol (SCTP).

UNIT-II

Application layer [3 Hrs]

Brief overview of protocols in Application Layer: Domain Name Systems, Hyper Text Transmission Protocol, TELecomunications NETwork (TELNET), File Transfer Protocol, Dynamic Host Configuration Protocol. E-mail: Architecture and services.

Network security [4 Hrs]

Principles, Symmetric and Asymmetric Cryptography, Confidentiality, Authenticity, Integrity and Non-repudiation. Symmetric key algorithms: Data Encryption Standard (DES), Public key algorithms: Rivest,

Shamir and Adleman algorithm (RSA).

Software Defined Network (SDN) [3 Hrs]

History and evolution of Software Defined Network, Control and data plane separation, Control Plane, Network Virtualization, Data Plane, Open Flow.

Fundamentals of Distributed Systems [5 Hrs]

Introduction and challenges of Distributed Systems, Systems models, Logical time and logical clocks, Global states, Externval Data Representation and Marshalling, Request-Reply protocols, Remote Procedure Call, Remote Method Invocation.

Text Books:

1. Andrew S. Tanenbaum, "Computer Networks", PHI.
2. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw-Hill.
3. George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems: Concepts and Design",Pearson
4. Pradeep K. Sinha, "Distributed Operating Systems", PHI.

Reference Books:

1. William Stallings, "Data and Computer Communications", PHI.
2. Alberto Leon-Garcia, Indra Widjaja, "Communication Networks – Fundamental Concepts and KeyArchitectures", Tata McGraw-Hill
3. Kurose Ross, "Computer Networks – A Top-Down Approach featuring the Internet", Pearson.
4. Thomas Nadeau and Ken Gray "SDN – Software Defined Networks"- O'Reilly Media
5. Patricia A Morreale and James M. Anderson, "Software Defined Networking: Design and Deployment"- CRC Press.
6. Mukesh Singhal, Niranjan G. Shivaratri, "Advanced Concepts in Operating System", McGraw Hill.

Department of Computer Science and Engineering

CS 16** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE-IV

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 16** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE-V

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 16** **(2L+1T hrs/week)**

PROGRAMME ELECTIVE-VI

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

C336 **CS 1663** **(3 hrs/week)**

COMPUTER NETWORK LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CS 1606 Computer Networks - II and associated prerequisites.

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

1. Identify and understand various functions used in socket programs
2. Develop and test of socket program for client server interaction for various purpose.
3. Implementing and Validate Sub network with static and various dynamic routing protocols.
4. Analyze the packet structure of various protocols used for communication
5. Understand the fundamentals of SDN

Department of Computer Science and Engineering

C352 **CS 1666** **(3 hrs/week)**

PARALLEL PROGRAMMING LAB

Objective: At least 10 experiments covering the syllabus to be carried out using programming skill of the subject concerned to get insight into the practical applications. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Concepts of Data Structures and Sorting Algorithms, Mutliprocessor, Multiprogramming architectures and their functioning.

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

1. Classify the different parallel programming constructs.
2. Simulate various parallel programming constructs in any high level language.
3. Select appropriate constructs to be used in different algorithms.
4. Justify the output of a program in solving different problems and evaluate its performance and effectiveness.
5. Compare the performance of programs designed on two different paradigms.

C407 **CS 1671**

MINI PROJECT

Objective: The students are required to undertake innovative and research oriented project under the direct supervision of a faculty member of the department. The mini project should not only to reflect their knowledge gained in the previous seven semesters but also to acquire additional knowledge and skill of their own effort. The Mini Project are assigned at the end of Vth semester and the final evaluation and grades are awarded at the end of VIth semester

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

1. Describe the basic concepts of computer science related to the problem under consideration
2. Identify an appropriate engineering problem to be solved
3. Construct an appropriate design methodology for software development and demonstrate effective communication and writing skills
4. Experiment on a designed model and develop an ability to work in a team
5. Evaluate the project based on application of knowledge and practical understanding of the model

Department of Computer Science and Engineering

C403 **CS 1703** **(3L + 1 T hrs/week)**

COMPILER DESIGN

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: The course is aimed at offering complete knowledge on compiler design and ends with the development of a working compiler in parts. Topics include compiler structure, symbol tables, regular expressions and languages, finite automata, lexical analysis, context-free languages, LL(1), recursive descent, LALR(1), and LR(1) parsing semantic analysis, and code generation. This will enable the learners to use formal attributed grammars for specifying the syntax and semantics of programming languages and their impact on compiler design.

Pre-requisites: Discrete Structures for Computer Science, Formal Language and Automata Theory and Programming skills.

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

1. Define the competence in designing compilers.
2. Describe the functioning in various problem domains.
3. Discover complex engineering problem and able to solve it using the concept of compiler design.
4. Analyze and select optimal design scheme using compiler design principles
5. Develop and experiment using knowledge of compilers for compiling different programs.

UNIT- I

Compiler Structure [4 Hrs]
Analysis-synthesis model of compilation, Various phases of a compiler, Tool based approach to compiler construction, Input Buffering.

Lexical Analysis [6 Hrs]
Interface with input, Parser and symbol table, Token, Lexeme and patterns, Regular definition, Transition diagrams, LEX.

Syntax Analysis [8 Hrs]
CFG's, Ambiguity, Associativity, Precedence, Top down parsing, Recursive descend parsing,Transformation on the grammars, Predictive parsing, Bottom up parsing, Operator precedence grammars, LR parsers (SLR, Canonical, LALR), YACC.

Syntax Directed Translation [2 Hrs]
Inherited and synthesized attributes, Dependency graph, Evaluation order, Bottom up evaluation of S-attributed definitions L- attributed definitions and top down translation of attributes.

UNIT-II

Type Checking [2 Hrs]
Type system, Type expressions, Structural and name equivalence of types, **Type conversion**.

Run Time Environments [4 Hrs]
Storage organization, Storage-allocation strategies, Access to nonlocal names, Activation tree, Activation record, Parameter passing, Symbol table and **dynamic storage allocation.**

Intermediate Code Generation [8 Hrs]
Intermediate representations, **Translation of declarations, Assignments, Control flow,** Boolean expressions and procedure calls ,Three-address code, Implementation of three address statements

(Quadruples, Triples, Indirect triples).

Code Generation [6 Hrs]

Issues in the design of a code generator, Basic blocks and flow graphs, Next use information, **Register allocation, Code generation algorithm,** Dag representation of programs, **Code generation from dags, Peephole optimization and code generator generators.**

Text Books:

1. A.V. Aho, R. Sethi, J.D. Ullman, "Compilers: Principles, Techniques and Tools", Addison – Wesley.
2. Steven S. Muchnick, "Advanced Compiler Design and Implementation", Elsevier.

Reference Books:

1. W. Appel, "Modern Compiler Implementation in C: Basic design", Cambridge Press.
2. Fraser and Hanson, "A Retargetable C Compiler: Design and Implementation", Addison-Wesley.
3. Dhamdhere, "Compiler Construction", McMillan.
4. A. V. Aho and J. D. Ullman, "Theory of Parsing, Translation and Compiling", Prentice Hall.

Department of Computer Science and Engineering

C404 **BA 1710** **(2L + 1 T hrs/week)**

INDUSTRIAL ENGINEERING MANAGEMENT

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

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

1. Understanding different concepts regarding Organization and Productivity in industries.
2. Evaluate control charts for variable as well as attribute data for different types of manufacturing processes and service activities.
3. Identify the techniques, skills and modern engineering tools necessary for engineering practice.
4. Applying knowledge of science, mathematics and engineering.
5. Analyze the concept of Inventory management, reliability.

UNIT-1

Philosophy and Development of Management thought:
Concept and definition of management, Functions and Roles of Management, Social Responsibilities of Management, Pioneers in Management, Taylor's Scientific Management, Contribution of Henry Fayol, Gilberth and Mayo.

Schools of Management Thought:
Human Behaviour, Social System, Systems approach, management process school. Control Charts for SQC : Statistical Quality Control (SQC). Control charts or variables such as X, R charts and control charts for attributes such as p-chart, c-chart.
Construction & use of the control charts. Process capability.

Reliability:
Introduction to reliability, bath-tub curve. Life expectancy. Reliability based esign. Series & Parallel System. **Defect Diagnosis and prevention : Basic causes of failure, curve/control of failure,** MTBF. Maintainability, **Condition monitoring and dignostic techniques.**

Value Engineering:
Elements of value analysis Techniques

UNIT-II

Quantitative Techniques in Managerial Decisions:
Concept of budget and budgetary control, Time-event network analysis; ABC Analysis, Break-even Analysis; Decision Tables; Concept of productivity, measuring productivity, Use information technology

Production Management:
Types of production; Types of Planning, Manufacturing Planning; Production planning, Scheduling; Work study & Method Study; Systems of wage payments, bonus, Automation. Organization of production, planning and control department.
Practice of purchasing and materials management, quality, quality standards and inspection, sources of supply; pricing principles and practices, Inventory Mangement, EOQ model.

Text Books:

1. H. Koontz and H. Weihrich, "Management", McGraw Hill, 1989.
2. Dobler W.D. "Purchasing & Materials Management", TMHC, New Delhi, 1984

Department of Computer Science and Engineering

CS 17** **(2L +1T hrs/week)**

PROGRAMME ELECTIVE-VII

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question
Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 17** **(2L +1T hrs/week)**

PROGRAMME ELECTIVE-VIII

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question
Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

CS 17** **(3L +1T hrs/week)**

OPEN ELECTIVE II

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question
Objectives and Pre-requisites are given under Elective details.
[TO BE CHOSEN FROM POOL OF ELECTIVES]

C406 **CS 1762** **(3 hrs/week)**

COMPILER DESIGN LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.
Pre-requisites: Corresponding theory paper CS 1703 Compiler Design and associated prerequisites.
Course Outcome: On successful completion of this course, students will be able to:

1. Simulate the functioning of various phases of compilers in various problem domains.
2. Devise solutions to solve complex engineering problem.
3. Identify an optimal design among all available design alternatives for further manipulation using compiler design principles to execute the code faster.
4. Apply tools using knowledge of compilers for compiling different programs.
5. Evaluate the effectiveness of different compilation tools.

Department of Computer Science and Engineering

CS1871#Minor Specialisation **(2L+1T hrs/week)**

DISTRIBUTED COMPUTING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: To understand the foundations of distributed systems. To learn issues related to clock Synchronization and the need for global state in distributed systems. To learn distributed mutual exclusion and deadlock detection algorithms. To understand the significance of agreement, fault tolerance and recovery protocols in Distributed Systems. To learn the characteristics of peer-to-peer and distributed shared memory systems

Pre-requisites: Basic concept of Computer Networks.
Course Outcomes: On successful completion of this course, students will be able to:

1. Understand the basics of distributed computing.
2. Observe distributed file system.
3. Construct distributed system support along with tansaction, security and replication in distributed computing.
4. Illustrate the ability to apply the knowledge in analysing and designing distributed systems.
5. Summarize the SOA and SOA security

UNIT I

Introduction to Distributed Computing [6]

Introduction to Distributed Systems, Examples Of Distributed Systems, Challenges, Architectural Models, Fundamental Models, Introduction to Interprocess Communications, External Data Representation and Marshalling Client Server Communication, Group Communication, Case Study: IPC In UNIX , Case Study: RMI, CORBA, Advances in Distributed Systems.

Distributed Objects and File System [6]

Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications, Java RMI Case Study, Introduction to DFS, File Service Architecture, Sun Network File System, Introduction to Name Services- Name Services and DNS, Directory and Directory Services, Distributed Mutex and Deadlock

Distributed Operating System Support [6]

The operating system layer, Protection- Process and Threads, Communication and Invocation, Operating System Architecture, Introduction to Time and Global States, Clocks, Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Distributed Debugging, Distributed Mutual Exclusion.

Transaction and Concurrency Control [4]

Transactions, Nested Transaction, Locks, Optimistic Concurrency Control, Timestamp Ordering, Comparison of Methods for Concurrency Control, Introduction to Distributed Transactions, Flat and Nested Distributed Transactions, Atomic Commit Protocols, Concurrency Control in Distributed Transactions, Distributed Deadlocks, Transaction Recovery.

UNIT II

Security and Replication [5]

Overview of Security Techniques, Cryptographic Algorithms, Digital Signatures, Cryptography Pragmatics, Replication, System Model and Group Communications, Fault Tolerant Services, Highly Available Services, Transactions with Replicated Data Issues in Designing Distributed System and Role of Middleware in Distributed System

Service-oriented architecture (SOA) [8]

Basic SOA Definition, Overview of SOA, SOA and Web Services, Service Oriented Grid, SOA Design and Development, Advantages and Future of SOA, SOA Support in J2EE, Java API for XML-based web services (JAX-WS), Java architecture for XML Binding (JAXB), Java API for XML Registries (JAXR), Java API for XML based RPC (JAX-RPC), Web Services Interoperability Technologies (WSIT), SOA support in .NET, Common Language Runtime, ASP.NET web forms, ASP.NET web services, Web Services Enhancements 08 (WSE).

SOA Security [6]

New Approach to Security for SOA, Extending SOAP for Security, Claiming and Verifying Identity with Passwords, WS-Security Standards, Kerberos with WS-Security, Encrypting SOAP Messages, XML Signatures, Implementing Security as a Service.

Text Books

1. Distributed O.S Concepts and Design , P.K.Sinha, PHI
2. Newcomer, Lomow, "Understanding SOA with Web Services", Pearson Education, 2005.
3. SOA Security, Ramarao, Manning.

Reference Books:

1. Advanced concepts in Operating Systems , Mukesh Singhal & N.G.Shivaratri, TMH
2. Distributed Computing , Sunita Mahajan, Seema Shah, OXFORD University Press
3. Distributed System Principles and Paradigms , Andrew S. Tanenbaum, 2nd edition , PHI
4. Distributed Systems , Colouris , 3rd Edition

Thomas Erl, "SOA Principles of Service Design "(The Prentice Hall Service-Oriented Computing Series from Thomas Erl), 2005.

Sharma

Signature
Date

Department of Computer Science and Engineering

CS 1872#Minor Specialisation **(2L + 1 Thrs/week)**

SOFT COMPUTING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: It comprises of computational techniques like Genetic/ Evolutionary algorithms, Artificial Neural Networks, Fuzzy Systems, Machine learning and probabilistic reasoning etc. This course thoroughly discusses Genetic Algorithms, Artificial Neural Networks (major topologies and learning algorithms) and Fuzzy Logic. At the end of the course, the students will be able to solve a variety of problems in their area of interest ranging from Optimization problems to Pattern recognition and Control tasks, by using soft computing tools.

Pre-requisites: Design and analysis of algorithms, Programming concepts.

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

1. Acquire knowledge of soft computing theories fundamentals and so they will be able to design program systems using approaches of these theories for solving various real-world problems.
2. Awake the importance of tolerance of imprecision and uncertainty for design of robust and low-cost intelligent machines.
3. Appreciate the importance of tolerance of imprecision and uncertainty for design of robust and low-cost intelligent systems.
4. Apply specified techniques in design and implementation of soft computing models for solving real life problems.
5. To acquire the knowledge of the fuzzy Neural network and Genetic Language.

UNIT-I

Introduction [4 Hrs]

Definition, Aspects of soft computing, Dealing with vagueness: Fuzzy systems, Rough sets, Modeling the brain-human cognition, Artificial neural networks, Modeling nature's optimization process: Natural evolution.

Fuzzy Set Theory [6 Hrs]

Review of crisp set theory: Sets and subsets, Definitions & notations - Universal set, Null set, Empty set, Subsets, Power set, Venn diagram, Operations on sets: Union, Intersection, Complementation, Difference Symmetric difference, Cartesian product, Properties of sets: Commutative, Associative, Distributive, DeMorgan's Law, Fuzzy sets: Fuzziness, Vagueness.

Fuzzy Membership Function [5 Hrs]

Crisp membership, Fuzzy membership, Membership profiles, Fuzzy sets: Definition, Notation, Features, Normality, Height, Support, Core cardinality, Transformation: Normalization, Dilation, Concentration, Contrast intensification, Fuzzification.

Fuzzy Set Operations [5 Hrs]

Operators – Union, Intersection, Complementation, Equality, Inclusion, Product, Difference, Disjunctive sum, Properties – Commutative, Associative, Distributive, Idempotency, De Morgan's law, Law Boundary conditions, Law of involution, Transitive law.

Department of Computer Science and Engineering

UNIT-II

Fuzzy Logic [5 Hrs]

Propositional logic: Propositions, Propositional logic well-formed formulae, Properties of wffs, Interpretation of logical expression, Logical equivalence, Tautology, Contradiction, Consistency, Validity of an argument.

Artificial Neural Networks [5 Hrs]

Basic concepts: The biological neuron, The artificial neuron, Characteristics of the brain, The McCulloch-Pitts neural model, The perceptron neural network architectures: Single layer feed forward ANNs, Multi-layer feed forward ANNs.

Back Propagation [5 Hrs]

Multilayer feed forward net- structure, Notations, Activation function, Generalized delta rule, The Back propagation Algorithm: Learning, Parameter choice, Initialization, Stopping criteria, Training set, Data representation, Hidden layers.

Advanced Search Algorithms [5 Hrs]

Genetic algorithms : Natural evolution, Chromosomes, Natural selection , Cross-over, Mutation, Basic GA, Encoding a solution as chromosome, decoding it, Fitness function, Population, GA operators- Selection, Tournament, Roulette wheel, Cross-over, Mutation, GA parameters , Convergence.

Text Books:

1. J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", Prentice Hall.
2. Melanie Mitchell, "An Introduction to Genetic Algorithms", PHI.

Reference Books:

1. Simon Haykin, "Neural Networks – A Comprehensive Foundation", Prentice Hall.
2. Jerry M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions", Prentice Hall
3. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", Wiley India.
4. Laurene Fausett, "Fundamentals of Neural Networks – Architecture, Algorithms and Applications", Pearson.

Bharma

Department of Computer Science and Engineering

C430

CS 1875
MAJOR PROJECT

The students are required to undertake innovative and research oriented project, not only to reflect their knowledge gained in the previous seven semesters but also to acquire additional knowledge and skill of their own effort. During their major project, the students are required to submit progress of their work in phases to make the department aware of his/her project. At the end of 16 weeks, students have to report to the internal guides/faculty members for final refinement and documentation. It is mandatory to follow the software engineering methodologies in carrying out the project work. The project is evaluated through internal presentation before the panel of faculty members followed by the evaluation by external examiner appointed by the university.

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

1. Apply modern engineering tools, techniques, and resources to solve different existing engineering problems.
2. Propose set of alternative design solutions to problems for which standard algorithmic solutions do not exist.
3. select proper Plan/manage an engineering activity within time.
4. Illustrate the able to comprehend technical literature and document project work.
5. Test the ability of listening, speaking, and presentation.

Department of Computer Science and Engineering

C228 **CS 1434/CS 1421** **(2L+1T hrs/week)**

JAVA PROGRAMMING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: The Java Programming Language course provides students with a solid foundation for programming with JAVA. It also highlights the creation of graphical user interfaces (GUIs), exceptions, file input/output (I/O), and threads; and network programming.

Pre-requisites: Object Oriented Programming.

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

1. Demonstrate an ability describe and recall the various object oriented concept
2. Formulate and interpret an object oriented model and design solution using object oriented concepts.
3. Ability to illustrate a diverse set of design solutions using techniques of Interface, Packages, File Handling, Multi-threading, etc.
4. Select and design methodology using bottom up approach and solve open ended problems using Java.
5. Define and create modern engineering tools, techniques and resources to build software using Java programming language.

UNIT-I

Introduction to Java [4 Hrs]
Evolution and features of java, Overview of java, **Two control statements, Lexical issues**, Data types, Variables and arrays, **Literals, Variables**, Type conversion and casting, Type promotion in expression, arrays, **Operators, Bitwise operators, Relational operators, Boolean and logical operators, Assignment Operators, The '?' operator, Operator precedence,** JAVA statements.

Introducing classes [2 Hrs]
Class fundamentals, Declaring objects, Assigning object reference Variables, Introducing methods, Constructors, 'this' keyword, **Garbage collection**, The finalize() method, **stack class.**

Methods and classes [4 Hrs]
Overloading methods and constructors, using object as parameters, Argument passing, Returning objects, **Recursion**, Access control, Static methods, Nested and inner classes, Command line argument.

Strings handling [4 Hrs]
String constructors, String length, Special string operators, Character extraction, String comparison, String searching, String modification, Changing case of characters within a string, Compression and String buffer, String builder.

Inheritance, Interfaces and Packages [4 Hrs]
Basics of inheritance, Types of inheritance, Using super keyword, method overriding, Dynamic method dispatch, Abstract class, Using final with inheritance, The object class, Defining and implementing interface, Extending interfaces, Nested interfaces, Applying interfaces, Defining and creating packages, Access protection, Importing packages.

Exception Handling [3 Hrs]
Fundamentals, Exception types, Uncaught exceptions, Using try and catch, Multiple catch clauses, Nested try statements, **Throws**, Finally, Java's built in exceptions, Creating own exception classes.

Department of Computer Science and Engineering

UNIT II

Input/Output and file handling [4 Hrs]
Java I/O classes and interfaces, The stream classes, Byte streams, The character streams, The console class, File class, Byte-stream class, Random access files.

Multithreaded programming [5 Hrs]
Thread basics, Java's thread model, Thread priorities, **Synchronization**, Messaging, Thread class and runnable interface. The main thread, Creating a thread, Creating multiple threads, Interthread communication, Suspending/resuming and stopping threads.

Network programming [5 Hrs]
Networking basics, The networking classes and interfaces, The InetAddress class, Inet4Address, TCP socket, **URL, URLConnection, HTTP/URL Connection, TCP/IP server sockets, Datagram socket and Datagram Packet.**

Event Based Programming [5 Hrs]
The applet class, Repaint(), The HTML applet tag, Passing Parameter to applet, Event handling, **Using delegation event model, Abstract Window program, Displaying information within a window, AWT controls.**

Text Books:

1. Programming With JAVA, 2nd Edition, E. Balaguruswami and TMH Publication.
2. Java: The Complete Reference, 7th Edition, Herbert Scheldt, TMH Publication.

Reference Books:

1. The Java Programming Language: K.Arnold and J. Gosling.
2. Professional java Server Programming: Allamaraju.
3. JAVA2: The Complete Reference, 3rd Edition, Patrick Naughton and HarbertSchildt, TMH Publication.
4. Internet & Java Program: R.Krishnamoorthy& S. Prabhu, New Age Internet Publisher.

Department of Computer Science and Engineering

C310 **CS 1532** **(2L +1T hrs/week)**

ADVANCED JAVA ROGRAMMING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: This course covers the advanced topics in java programming such as collection framework, language package, Network Programming, GUI programming using AWT and Swings, advanced Web Programming using Servlet and JSP, and Accessing Database with Java.

Pre-requisites: Object Oriented programming, Database Management System, Computer Network and Distributed System.

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

1. Write program codes displaying competence in basic object-oriented programming using Java
2. Compare and able to develop scalable network applications using Java
3. Illustrate and practice component-based software.
4. Explain and able to conduct investigations of technical issues consistent with their level of knowledge on distributed applications.
5. Design and implement the understanding pertaining to database applications

UNIT- I

Java fundamentals [4 Hrs]
Java I/O streaming, Filter and pipe streams, **Byte code interpretation**, **Threading**, Swing.

Network programming in java [8 Hrs]
Sockets, secure sockets, custom sockets, UDP datagrams, Multicast sockets, URL classes, Reading data from the server, Writing data, Configuring the connection, Reading the header, **Telnet application**, Java messaging services.

Applications in distributed environment [10 Hrs]
Remote method invocation, activation models, RMI custom sockets, Object serialization, RMI, **IIOP implementation, CORBA, IDL technology, Naming services, CORBA programming models**, JAR file creation.

UNIT-II

Multi-tier application development [10 Hrs]
Server side programming, Servlets, Java server pages, Applet to applet communication, Applet to servlet communication, JDBC, Applications on databases, **Multimedia streaming applications**, **Java media framework**.

Enterprise applications [8 Hrs]
Server Side Component Architecture, Introduction to J2EE, Session Beans, Entity beans, **Persistent entity beans**.

Text Books:

1. Elliotte Rusty Harold, "Java Network Programming", Shroff.
2. Ed Roman, "Mastering Enterprise Java Beans", John Wiley & Sons.

Reference Books:

1. Patrick Naughton, "Complete Reference: Java2", Tata McGraw Hill.
2. Hortsmann& Cornell, "Core Java 2 Advanced Features, VOL II", Pearson education.

3. Amir Afzal, "Advanced Java Programming", Pearson.
4. Joe Wigglesworth, Paula McMillan, "Java Programming: Advanced Topics", Course Technology.

Department of Computer Science and Engineering

C304 **CS 1534 /CS 1507** **(2L + 1 T hrs/week)**

DISCRETE STRUCTURE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course emphasizes on mathematical structures for describing data, algorithms and computing machines. Theory and Applications of sets, relations, functions, combinatorics, matrices, graphs and algebraic structures, which are pertinent to computer science are also covered.

Pre-requisites: Data Structures and Concepts of algorithms. Some programming experience is helpful but not necessary.

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

1. Identify effectively algebraic techniques to analyze basic discrete structures and algorithms.
2. Apply the concepts of sets, integers, reals and functions of such quantities to solve simple problems
3. Understand and relate the graphs and related discrete structures to practical examples
4. Understand the notion of mathematical and algorithmic thinking and apply them for problem solving
5. Infer logical reasoning to solve a problem

UNIT-I

Sets [2 Hrs]
Definition, **Operations, Counting, Comparing the size of sets, Countable sets**, Diagonalization, Limits of computability, Bags (multisets).

Ordered structures [3 Hrs]
Tuples, Lists, Strings and languages, Relations.

Inductively defined sets [5 Hrs]
Numbers, Strings, Lists, Binary trees, Cartesian products of sets.

Recursive functions and procedures [6 Hrs]
Numbers, Strings, Lists, Binary trees, Infinite sequences, Recursion in programs, Repetitive program design.

Propositional calculus [4 Hrs]
Well-formed formulas and semantics, Equivalence, Truth functions and normal forms.

UNIT-II

Predicate logic [5 Hrs]
Predicates, Qualifiers, Well-formed formulas, Semantics and interpretations, Validity, Equivalence, Normal Forms, Formalizing English sentences.

Program logic [5 Hrs]
Equality, Imperative program correctness, **Array assignment**, Termination.

Automatic reasoning [5 Hrs]
Clauses, Propositions, Substitution and unification, Resolution, Logic programming: Family trees, Logic program, Logic programming techniques.

Algebraic structures and abstract data types [5 Hrs]

Natural numbers, Lists and strings, Stacks and queues, **Binary trees** and Priority queues, **Abstract arrays, Container classes**.

Text Books:

1. James L. Hein, "Discrete Structures, Logic and Computability", Narosa.
2. J.P. Tremblay, R.Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill.

Reference Books:

1. Leon S. Levy, "Eastern, Discrete Structures of Computer Science", Wiley
2. C.L.Liu, "Elements of Discrete Mathematics", McGraw-Hill.
3. Bruce Mills, "Theoretical Introduction to Programming" , Springer.
4. Fletcher R. Norris, "Discrete structures: An introduction to mathematics for computer science" Prentice Hall.

Department of Computer Science and Engineering

C316 **CS 1538/ CS 1601** **(2L + 1 T hrs/week)**

OBJECT ORIENTED ANALYSIS AND DESIGN USING UML

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course delves into the processes of both object-oriented analysis and object-oriented design using UML as the notation language to provide a common, standard notation for recording both analysis models and design artifacts. Facets of the Unified Process approach to designing and building a software system are also covered.

Pre-requisites: Object oriented Design concepts, Design & Analysis of Algorithms and software engineering.

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

1. Identify object oriented design techniques suitable for a complex problem
2. Demonstrate an ability to formulate a solution plan and methodology for an engineering problem using object oriented analysis and design using UML.
3. Determination of the formula and interpretation of a model for project management.
4. Investigate a problem to identify technical issues and solve the problems using various object oriented design tools and techniques.
5. Design real-life application using modern UML tools, techniques and resources.

UNIT I

Introduction [2 Hrs]
Challenges in Software Engineering, Complexity of Software, Structure and Attributes of a Complex System, Evolution of Object Models - Programming Languages and Paradigms , Foundations of the Object Model - OOA, OOD and OOP, **About Object Orientated Technology,** Development and OO Modeling History.

Modeling Concepts [4 Hrs]
Elements of Object Model-Abstraction and Encapsulation, Modularity and Hierarchy, Typing, Concurrency and Persistence, Nature of an object: State, Behavior and Identity, Relationships among objects, **Modeling design Technique,** Three models- Class Model, State model and Interaction model. Overview of UML

Class Modeling [6 Hrs]
Nature of a class: Interface and Implementation, Relationships among classes, Object and class concepts, link and association, **Generalization and Inheritance,** Advanced class modeling- aggregation, Abstract class metadata, constraints. How to build quality Classes and Objects.

State Modeling [6 Hrs]
Event, state, Transition and conditions, state diagram, state diagram behavior, concurrency, Relation of Class and State models.

Interaction Modeling [2 Hrs]
Notations, Relationships and Examples - Use case Models, sequence models, activity models.

UNIT II

Analysis and Design [7 Hrs]
Development Life cycle, Development stages, Domain Analysis-Domain class model, domain state model, domain interaction model, Iterating and analysis. Application Interaction model, Application class model, Application state Model, Adding operation.

System Design [7 Hrs]

Estimating Performance, **Making a reuse plan**, breaking system into subsystems ,identifying concurrency, allocation of subsystems, management of data storage, Handling Global resources, choosing a software control strategy, Handling boundary condition, common Architectural style.

Class design [6 Hrs]

Overview of class design, designing algorithms, recursing downward, refactoring, design optimization, **Adjustment of Inheritance, Reification of Behavior.**

Text Books:

1. Michael R Blaha, James R Rumbaugh, "Object-Oriented Modeling and Design with UML", Pearson.
2. Ali Bahrami, "Object Oriented Systems using the United Modeling Language", McGraw Hill.

Reference Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language", Pearson Education.
2. Grady Booch, "Object Oriented Analysis and Design", Pearson Education.
3. Graig Larman, "Applying UML and Patterns", Addison Wesley.
4. Perdita Stevens, Rob Pooley, "Using UML Software Engineering with Objects and Components",Pearson.

Signature
Date

Department of Computer Science and Engineering

C345 **CS 1645/CS 1605** **(2L + 1 T hrs/week)**

UNIX INTERNALS AND SHELL PROGRAMMING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objective: This course focuses on the principles and techniques of UNIX Operating System's concepts and terminologies, including file system programming and shell programming which includes advance UNIX commands and utilities, process management, signal management, Inter-process Communication issues and techniques.

Pre-requisites: Data Structures, Operating Systems and Programming language concepts. This course assumes that the student is familiar with C language and has some exposure to program writing in C language.

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

1. Describe the function of the basic UNIX commands
2. Identify various useful UNIX commands on a standard UNIX based Operating System
3. Write shell programming on UNIX based Operating System
4. Select suitable system calls for file handling
5. Choose appropriate algorithms for process control and synchronization

UNIT I

Introduction [4 Hrs]
Evolution of UNIX Operating System, Introduction to Multi-user System, Features and benefits of UNIX, Versions of UNIX, System Structure of UNIX, User perspective services, Operating System perspective services, Assumptions about hardware, Kernel architecture of UNIX Operating System.

Introduction to Shell scripts and Awk programming [6 Hrs]
Bourne Shell, C Shell, Shell Variables, Scripts, Meta characters and environment, if and case Statements, for, while and until Loops, Awk pattern scanning and processing, begin and end Patterns, Awk arithmetic and variables, Built-in functions and operators, Arrays, Strings.

The Buffer Cache and Internal Representation of Files [6 Hrs]
Buffer Headers, Structure of Buffer Pool, Scenarios for Retrieval of Buffer, Reading and writing Disk Blocks, Advantages and disadvantages of Buffer Cache, System calls for file system, Inodes, Structure of a regular file, Directories, Conversions of a pathname to an Inode, Super Block, Inode assignment to a new file, **Allocation of Disk Blocks.**

System calls for file systems [4 Hrs]
Open, Close, File creation, Special files creation, Read, Write, File and record locking, Adjusting the position of file I/O, lseek, Change Directory, Change Root, Change Owner and Change Mode, stat, fstat, Pipes, Dup and Dup2, **Mounting and un-mounting the file system, Link, Unlink, File System abstraction and maintenance.**

UNIT II

The Structure of Processes and Process control [6 Hrs]
Process States and Transitions, Layout of system memory, The Context of a Process, Manipulation of Process address space, Sleep Process creation and termination, User-ID of a Process, Changing the size of a Process, The Shell, System Boot and INIT process.

Process scheduling and memory management policies [6 Hrs]
Process scheduling with round robin multilevel feedback scheduler, Memory management policies, **Swapping, Demand Paging, Data structures for Demand Paging,** Page Stealer Process, Page faults, **Hybrid system with Swapping and Demand Paging.**

I/O Sub-systems and Inter-process Communications [8 Hrs]
Driver interfaces, Disk Drivers, Terminal Drivers, Streams, Process tracing, **System V Inter-process Communication, Network Communications and Sockets.**

Text Books:

1. Bach, Maurice J., "The Design of the Unix Operating System", PHI, 2004.
2. Karee Christian, "The Unix Operating System", John Wiley & Sons.

Reference Books:

1. Vahalia, "UNIX Internals: The New Frontiers", Pearson Education Inc, 2003.
2. Uresh Vahalia, "UNIX Internals: The New Frontiers", Prentice Hall, 2000.
3. M. Beck, et.al, "Linux Kernel Programming", Pearson Education Asia, 2002.
4. Sumitabha Das, "UNIX Concepts and Applications", McGraw Hill.

Department of Computer Science and Engineering

C347 **CS 1651** **(2L +1T hrs/week)**

LATEST TRENDS IN COMPUTER SCIENCE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: This course provides students with:

- Introduction to deep learning, a branch of machine learning concerned with the development and application of modern neural networks.
- An opportunity to explore the research issues in augmented Reality and Virtual Reality (AR &VR).
- It also acquaints the learners with basic concept and framework of virtual reality.
- It addressed to overview of intelligent robotics systems including the algorithms for the analysis of the data obtained by vision and range sensors, basic principles of modeling kinematics and dynamics and design of basic control strategies.

Pre-requisites: Design and Analysis of Algorithms, Basics of Soft computing and AI will be preferable.

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

1. Discuss various recent technologies used in computer science associated with research and application.
2. Select optimal design scheme suitable for data analytics techniques and tools for extraction of knowledge
3. Identify modern engineering tools, techniques and resources to solve various Computer Vision, Big Data, Block chain Problems, etc
4. Classify machine learning algorithms for sovling Artificial Intelligence problems or other related problems.
5. Decide the individual specialization from the recent trends in computer science of their interest.

UNIT I

Data Science [5 Hrs]
Introduction, Data Science project lifecycle, Statistics, Data Preparation, Model Building, Machine learning models, Performance measurement of a model, Communicating results, Python for Data Science.

Internet of Things (IoT) [5 Hrs]
Introduction to IoT, IoT Technologies and Architectures: Devices and Gateways, Local and wide area networking, Data management, Applications of Internet of Things, Opportunities and Challenges in the Internet of Things.

Computer Vision [6 Hrs]
Fundamentals of vision: Image Formation and Representation – Intensity and Range Images– Camera models – Camera Parameters – Light and color – Image noise – Image filtering – Image smoothing-Sharpening, Image features: point and line detection – Hough Transform – Edge detection – corner detection – Harris Detecter – Textures – Principal component analysis – feature descriptors – SIFT and SURF, High level vision: Geometric methods – Model based vision – Obtaining hypothesis by pose consistency – pose clustering using invariants – verification, Linear discriminant based classifiers and tree classifiers.

Machine Learning [5 Hrs]
Learning- Supervised and Unsupervised learning, adaptive Learning, Reinforcement learning, Linear classification, Loss minimization, Stochastic gradient descent, K-Means Algorithm, The perceptron. Learning by gradient descent. Multilayer perceptron and the back propagation algorithm, Deep learning, Auto-encoders, CNNs, RNNs, Introduction to Natural Language Processing.

UNIT II

Artificial Intelligence [6 Hrs]

Overview of Artificial intelligence, Problems and techniques, Problem solving agents, searching for solutions; various uniform search strategies, Introduction to Heuristic search strategies, Hidden Markov models, Learning Bayesian networks, Fuzzy sets & fuzzy logics

Block Chain Coding [6 Hrs]

Introduction: trustless system, Why blockchain, Decentralized transactions. History: How and when blockchain/bitcoin started, Milestones on the development of bit coin: creation, exchanges. Overview of blockchain technology: What is blockchain, Transactions: Recording transactions, Digital signature, Verifying and confirming transactions; Blocks, Hashes, Consensus: Distributed consensus, Byzantine generals problem, Proof of work; Verify and confirm blocks. Blockchain applications : Government, Identity management, Auto executing contracts, Three signature escrow, Triple entry accounting, Elections and voting?

Augmented Reality[7 Hrs]

Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality. What is Augmented Reality? History of Augmented Reality. Augmented and Mixed Reality, Mixed Reality. Mixed Reality continuum, Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications.

Text Books:

1. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly Media Publication
2. Bernd Jahne and Horst HauBecker, Computer vision and Applications, Academic Press, 2000
3. David A. Forsyth & Jean Ponce, Computer vision-A modern Approach, Prentice Hall, 2002
4. Introductory techniques for 3D computer vision, Prentice Hall, 1998
5. Arvind Narayanan, Joseph Bonneau, Edward Felten, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", Kindle Edition.
6. Andreas M. Antonopoulo,"Mastering Bitcoin: Programming the Open Blockchain",2nd Edition, Kindle Edition.

Reference Books

1. Andries P. Engelbrecht , "Computational Intelligence: An Introduction", Wiley 2nd Edition, 2007
2. Robin R. Murphy, "Introduction to AI Robotics", MIT Press, 2000
3. Leandro N. de Castro and Jonathan Timmis, "Artificial Immune system: A new Computational Intelligence Approach", Springer-Verlag, Germany 2002.

Department of Computer Science and Engineering

C349 **CS 1655/CS 1624** **(2L +1T hrs/week)**

AUTONOMOUS MOBILE ROBOTICS AND COMPUTATIONAL INTELLIGENCE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question.

Objectives: The course will cover basic principles of design and practice of intelligent robotics systems including the algorithms for the analysis of the data obtained by vision and range sensors, basic principles of modeling kinematics and dynamics and design of basic control strategies. Emphasize will also be given on formulating interesting robotics tasks and show how they can be accomplished by individual robot or cooperative robot teams (such as flocking, foraging as well as robotic soccer).

Pre-requisites: Design and Analysis of Algorithms, Basic Mathematics (linear algebra, calculus and probability), Soft computing.

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

1. Identify application of Robots
2. Describe the evolution, kind of robots and basics of design of a Robotics System
3. Choose correct choices of Locomotion, Kinematics, Perception techniques
4. Apply Intelligent Algorithms for decision making in Motion Planning and path optimization
5. Develop simple robot control systems integrating perception, planning, and action

UNIT-I

Introduction [4 Hrs]
Definition, Applications of mobile robotics, History of mobile robotics.

Design of system and navigation architecture [7 Hrs]
Reference control scheme of a mobile robotics environment, Temporal decomposition of architecture, Control decomposition, Hybrid architecture, Mobile architecture, Perception, **Representation and the mapping process**.

Locomotion [6 Hrs]
Issues for locomotion, Legged mobile robots, Wheeled mobile robots.

Kinematics [6 Hrs]
Kinematics introduction, Forward and reverse kinematics, Wheeled kinematics and its constraints, Mobile system locomotion, Human biped locomotion as a rolling polygon, **Representation of robot position through the reference frame.**

UNIT-II

Perception [4 Hrs]
Sensors for mobile robots, Sensor classification, Characterization and sensor performance, Wheeled motor sensor, Ground bases beacon, **Active ranging, Motion/Speed sensor, Vision based sensors.**

Navigation [3 Hrs]
Localization overview, Path planning.

Computational intelligence [5 Hrs]
Swarm intelligence, Evolutionary computation, **Artificial immune system, Ant algorithm.**

Mobile robot programming [7 Hrs]
This chapter is included to provide hands on introduction to the field of mobile robotics and various issues in designing and planning of robot work environment. It includes construction and programming of robotic agents using robotic kits and microcontrollers applying concepts of locomotion, perception, navigation and computational intelligent algorithms.

Text Books:

1. Ronald Siegwart, Illah R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", MIT Press,
2. Andries P. Engelbrecht , "Computational Intelligence: An Introduction", Wiley 2nd Edition, 2007

Reference Books:

1. Ronald C. Arkin , "Intelligent Robotics and Autonomous Agents", MIT Press, 1997
2. Ulrich Nehmzow, "Mobile Robotics: A practical Introduction", Springer-Verlag London, 2003
3. Robin R. Murphy, "Introduction to AI Robotics", MIT Press, 2000
4. Leandro N. de Castro and Jonathan Timmis, "Artificial Immune system: A new Computational Intelligence Approach", Springer-Verlag, Germany 2002.

Department of Computer Science and Engineering

C417 **CS 1742** **(2L +1T hrs/week)**

DATA ANALYTICS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective: Data Analytics is the science of analyzing data in order to come up with some decision making useful knowledge. The course presents wide range of data analytics techniques and tools for extraction of knowledge.

Pre-requisites: Database Systems.

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

1. Apply data analytics techniques and tools to produce knowledge from a given dataset.
2. Analyze and apply statistical and probabilistic approach for the development of a model using large dataset.
3. Create and evaluate the model using suitable machine learning algorithms.
4. Formulate statistical hypothesis testing and inference for the model development.
5. Explain the dataset using different data representation tools.

UNIT I

Introduction [4 Hrs]
Introduction to data analytics (DA), data preparation, and data cleaning, Big Data Overview, What is data sciences, The rising and importance of data sciences, **Big data analytics in industry**.

Data Analytics Lifecycle and methodology [4 Hrs]
Understanding Business Data, Data Preparation, Data Modelling, Data Evaluation, Communicating results, **Deployment of Data**.

Statistical Analysis [6 Hrs]
Basic statistical concepts. Mean, standard deviation. **Rank statistics and percentiles**, Distributions, Covariance, correlation, analysis of variance, Statistical tests, confidence and hypothesis testing, Tools such as R.

Probabilistic Analysis [6 Hrs]
Dependence and Independence, Conditional Probability, Bayes's Theorem, **Random Variables** Continuous Distributions, The Normal Distribution, The Central Limit Theorem.

UNIT II

Data Analytics: Theory & Methods [6 Hrs]
Data features, Classification, Supervised and unsupervised learning, Supervised learning - Linear/Logistic regression, Decision trees, Naïve Bayes, Unsupervised learning - K-means clustering, Association rules, Clustering algorithms, **Knowledge discovery**, **Anomaly detection**.

Hypothesis and Inference [4 Hrs]
Statistical Hypothesis Testing, Example: Flipping a Coin, Confidence Intervals, **P-hacking**, Example: Running an A/B Test, **Bayesian Inference**.

Tools for Data Analytics [4 Hrs]
Globally distributed data stores, Tools for big data, Introduction to Hadoop, HDFS, MapReduce, **YARN**, HBase, Hive, Pig, **Sqoop**, **Zookeeper**, **Flume**, NoSQL.

Data Representation and Analysis [6 Hrs]

Log Data Analysis – HDFS scenario: Write once & Read often, **Data Warehouse**, Fraud Detection, Risk Modeling, Social Sentiment Analysis, Image Classification, **Graph Analysis**.

Text Books:

1. Daniel T. Larose & Chantal D. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining", Wiley 2nd Edition.
2. Ron Klimberg and B. D. McCullough, "Fundamentals of Predictive Analytics with JMP" SAS Institute Publishers, 2013.

Reference:

1. EMC Education Services, "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data" Wiley Publications.
2. Joel Grus, "Data Science from Scratch First Principles with Python", O'Reilly Media, 2015.

Department of Computer Science and Engineering

C368 **CS 1720/CS 1626** **(2L+1T hrs/week)**

INTELLECTUAL PROPERTY RIGHTS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objective:

- Introduce fundamental aspects of Intellectual Property Rights to students who are going to play a major role in development and management of innovative projects in industries. The course introduces all aspects of the IPR Acts.
- Case studies to demonstrate the application of the legal concepts in Science, Engineering, Technology and Creative Design.
- The course is designed for raising awareness of a multidisciplinary audience and has been categorized under 'General'.

Pre-requisites: NIL.
Course Outcomes: On successful completion of this course, students will be able to:

1. Identify regulations, legislation and standards for Intellectual Property Rights
2. Describe the impact of Intellectual Property Rights on engineering and industrial practices vis a vis social, environmental and economic context
3. Apply principles of Intellectual Property Rights to sustainable design and development.
4. Analyze ethical lapses and recognize ethical dilemmas.
5. Distinguish professional issues which arise in the intellectual property law context

UNIT – I

Overview Of Intellectual Property [5 Hrs]
Introduction and the need for intellectual property right [IPR], IPR in India – Genesis and Development IPR in abroad, some important examples of IPR.

Patents [11 Hrs]
Macro-economic impact of the patent system, Patent and kind of inventions protected by a patent, Patent document, How to protect your inventions? , Granting of patent, Rights of a patent, How extensive is patent protection? , Why protect inventions by patents?, Searching a patent, Drafting of a patent, Filing of a patent, Case studies, **The different layers of the international patent system, [national, regional and international options], Utility models : Differences between a utility model and a patent?, Trade secrets and know-how agreements**

Copyright [4 Hrs]
What is copyright?, What is covered by copyright?, How long does copyright last?, Why protect copyright?, What are related rights? , Distinction between related rights and copyright?, Rights covered by copyright?

UNIT – II

TradeMarks [4 Hrs]
What is a trademark?, Rights of trademark?, What kind of signs can be used as trademarks? Types of trademark, function does a trademark perform, How is a trademark protected? How is a trademark registered?, How long is a registered trademark protected for ? How extensive is trademark protection?, **What are well-known marks and how are they protected?, Domain name and how does it relate to trademarks?**

Geographical Indications [2 Hrs]

What is a geographical indication?, **How is a geographical indication protected?, Why protect geographical indications?**

Industrial Designs [2 Hrs]

What is an industrial design?, How can industrial designs be protected?, What kind of protection is provided by industrial designs?, How long does the protection last?, Why protect industrial designs?

New Plant Varieties [2 Hrs]

Why protect new varieties of plants?, How can new plants be protected?, What protection does the breeder get?, How long do the breeder's rights last?, How extensive is plant variety protection?

Unfair Competition and Enforcement Of Intellectual Property Rights [5 Hr]

What is unfair competition?, relationship between unfair competition and intellectual property laws? Infringement of intellectual property rights, Enforcement Measures.

Intellectual Property [5 Hrs]

Overview of Biotechnology and Intellectual Property, Biotechnology Research and Intellectual Property Rights, **Management Licensing and Enforcing Intellectual Property Commercializing Biotechnology Invention Case studies of Biotechnology.**

Text Books

1. T. M Murray and M.J. Mehlman, Encyclopedia of Ethical, Legal and Policy issues in Biotechnology, John Wiley & Sons 2000.

References

1. P.N. Cheremisinoff, R.P. Ouellette and R.M. Bartholomew, Biotechnology Applications and Research, Technomic Publishing Co., Inc. USA, 1985.
2. D. Balasubramaniam, C.F.A. Bryce, K. Dharmalingam, J. Green and K. Jayaraman, Concepts in Biotechnology, University Press [Orient Longman Ltd.], 2002.
3. Bourgagaize, Jewell and Buiser, Biotechnology: Demystifying the Concepts, Wesley Longman, USA, 2000.
4. Ajit Parulekar and Sarita D' Souza, Indian Patents Law – Legal & Business Implications; Macmillan India ltd , 2006.

Department of Computer Science and Engineering

C402 **CS 1660/CS 1702** **(2L + 1 T hrs/week)**

SOFT COMPUTING

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objectives: It comprises of computational techniques like Genetic/ Evolutionary algorithms, Artificial Neural Networks, Fuzzy Systems, Machine learning and probabilistic reasoning etc. This course thoroughly discusses Genetic Algorithms, Artificial Neural Networks (major topologies and learning algorithms) and Fuzzy Logic. At the end of the course, the students will be able to solve a variety of problems in their area of interest ranging from Optimization problems to Pattern recognition and Control tasks, by using soft computing tools.

Pre-requisites: Design and Analysis of Algorithms, Programming concepts.
Course Outcomes: On successful completion of this course, students will be able to:

1. Differentiate between basic soft and hard computing models.
2. Analyze, evaluate and build fuzzy models.
3. Appreciate the importance of tolerance of imprecision and uncertainty for design of robust and low-cost intelligent systems.
4. Apply specified techniques in design and implementation of soft computing models for solving real life problems.
5. To acquire the knowledge of the fuzzy Neural network and Genetic Language.

UNIT-I

Introduction [4 Hrs]
Definition, Aspects of soft computing, Dealing with vagueness: Fuzzy systems, Rough sets, Modeling the brain-human cognition, Artificial neural networks, Modeling nature's optimization process: Natural evolution.

Fuzzy set theory [6 Hrs]
Review of crisp set theory: Sets and subsets, Definitions & notations - Universal set, Null set, Empty set, Subsets, Power set, Venn diagram, Operations on sets: Union, Intersection, Complementation, Difference Symmetric difference, Cartesian product, Properties of sets: Commutative, Associative, Distributive, DeMorgan's Law, Fuzzy sets: Fuzziness, Vagueness.

Fuzzy membership function [5 Hrs]
Crisp membership, Fuzzy membership, Membership profiles, Fuzzy sets: Definition, Notation, Features, Normality, Height, Support, Core cardinality, Transformation: Normalization, Dilation, Concentration, Contrast intensification, Fuzzification.

Fuzzy set operations [5 Hrs]
Operators – Union, Intersection, Complementation, Equality, Inclusion, Product, Difference, Disjunctive sum, Properties – Commutative, Associative, Distributive, Idempotency, De Morgan's law, Law Boundary conditions, Law of involution, Transitive law.

UNIT-II

Fuzzy logic [5 Hrs]
Propositional logic: Propositions, Propositional logic well-formed formulae, Properties of wffs, Interpretation of logical expression, Logical equivalence, Tautology, Contradiction, Consistency, Validity of an argument.

Artificial neural networks [5 Hrs]
Basic concepts: The biological neuron, The artificial neuron, Characteristics of the brain, The McCulloch-

Pitts neural model, The perceptron neural network architectures: Single layer feed forward ANNs, Multi-layer feed forward ANNs.

Back propagation [5 Hrs]

Multilayer feed forward net- structure, Notations, Activation function, Generalized delta rule, The Back propagation Algorithm: Learning, Parameter choice, Initialization, Stopping criteria, Training set, Data representation, Hidden layers.

Advanced search algorithms [5 Hrs]

Genetic algorithms : Natural evolution, Chromosomes, Natural selection , Cross-over, Mutation, Basic GA, Encoding a solution as chromosome, decoding it, Fitness function, Population, GA operators-Selection, Tournament, Roulette wheel, Cross-over, Mutation, GA parameters , Convergence.

Text Books:

1. J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", Prentice Hall.
2. Melanie Mitchell, "An Introduction to Genetic Algorithms", PHI.

Reference Books:

1. Simon Haykin, "Neural Networks – A Comprehensive Foundation", Prentice Hall.
2. Jerry M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions", Prentice Hall
3. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", Wiley India.
4. Laurene Fausett, "Fundamentals of Neural Networks – Architecture, Algorithms and Applications", Pearson.

Department of Computer Science and Engineering

C423 **CS 1728 / CS 1701** **(2L + 1 T hrs/week)**

DISTRIBUTED SYSTEMS

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objectives: This course is designed to present the concepts of heterogeneous multi-computer systems and distributed operating systems. Communication in a client/server model using RPC, Message oriented communications, remote object invocation, and distributed processes and software agents are discussed. Other distributed systems concepts such as clock synchronization, data consistency and replication, fault tolerance, security and distributed component and file systems are also covered.

Pre-requisites: Operating Systems, Database Management Systems, Computer Networks – I, Computer Networks – II.

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

1. Observe the complexities of distributed system development.
2. Develop a strategy to overcome important issues including time, inter-process communication, and state management in distributed computing.
3. Construct systems in a constrained distributed environment by integrating several modules and validate the detailed design.
4. Illustrate the ability to apply the knowledge in analysing and designing distributed systems.
5. Summarize the middleware technologies that support distributed applications such as RPC, RMI and object-based middleware

UNIT – I

Fundamentals [5 Hrs]
Introduction to Distributed Systems, Evolution and applications of DCS, Challenges and examples of distributed systems, Distributed systems models: Architectural models and fundamental models.

Time and Global States [5 Hrs]
Logical time and event ordering, Global state and snapshot algorithms, clock synchronization.

Coordination and Agreement [6 Hrs]
Introduction, Distributed mutual exclusion, distributed shared memory: General architecture, **Design and implementation issues**, elections.

Distributed Transactions [4 Hrs]
Flat and nested distributed transactions, Atomic commit protocol, **Concurrency control in distributed transactions**, Distributed deadlocks and recovery.

UNIT – II

Fault Tolerance and Load Balancing [4 Hrs]
Fault tolerance and recovery: Fault models, Fault tolerant services, reliable communication, Resource sharing and load balancing.

Inter-Process Communication [4 Hrs]
API for the Internet protocols, external data representation and marshalling, Multicast communication, **Indirect Communication: Publish-subscribe systems.**

Remote Invocation [5 Hrs]
Request-reply protocols, Remote procedure call, Remote method invocation, **Case study: Java RMI.**

CORBA RMI.

Security and Authentication [3 Hrs]
Overview of security techniques, Digital signatures, Kerberos.

Distributed Files Systems [4 Hrs]
Features of DFS, File service architecture, **Case study: Sun Network File System, Andrew File System, Google file systems (GFS).**

Text Books:

1. George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems: Concepts and Design",Pearson
2. Andrew S. Tanenbaum, Maarten van Steen, "Distributed Systems: Principles and Paradigms", Pearson

Reference Books:

1. Pradeep K. Sinha, "Distributed Operating Systems", PHI.
2. Nancy A. Lynch, "Distributed Algorithms", Elsevier.
3. Kenneth P. Birman, "Reliable distributed systems: Technologies, Web services, and applications",Springer.
4. Paulo Veríssimo, Luis Rodrigues, "Distributed systems for system architects", Springer.

Department of Computer Science and Engineering

C409 **CS 1721/ CS1407 (2L + 1 T hrs/week)**

PRINCIPLES OF PROGRAMMING LANGUAGE

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objectives: The objective of this course is to identify the conceptual building blocks from which languages are assembled and specify the semantics, including common type systems, of programming languages.

Pre-requisites: Programming languages & Concepts.
Course Outcomes: On successful completion of this course, students will be able to:

1. Examine the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
2. Classify notations to describe syntax and semantics of programming languages.
3. Analyse the behaviour of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
4. Classification of the concepts of ADT and object oriented programming for large scale software development.
5. Evaluate the concepts of concurrency control and exception handling in programming language

UNIT-I

Introduction: Names, Scopes, and Bindings [7 Hrs]
The art of language design; Programming language spectrum; Why study programming languages? **Evolution of programming languages – describing syntax** – context-free grammars – attribute grammars – describing semantics – lexical analysis – parsing – recursive-decent – bottom up parsing. Syntactic Structure - Language representation, Abstract Syntax tree, Lexical syntax, Context Free Grammars, Variants of CFG, Issues involved and Normal Forms for CFG. Compilation and interpretation; Programming environments. Names, scope, and bindings: The notion of binding time; Object lifetime and storage management; Scope rules; Implementing scope; The meaning of names within a scope; The binding of referencing environments; Macro expansion. **High Level Languages, Issues in Programming - Case studies, Programming paradigms, Language implementation.**

Data Types & Control Structures [6 Hrs]
Type systems; Type checking; Records and variants; **Arrays; Strings; Sets; Pointers and recursive types; Lists; Files and Input/Output;** Equality testing and assignment. Names – variables – binding – type checking – scope – scope rules – lifetime and garbage collection – primitive data types – strings – array types – associative arrays – record types –union types – pointers and references – Arithmetic expressions – overloaded operators –type conversions – relational and boolean expressions – assignment statements – mixed mode assignments . Control structures – selection – iterations – branching – guarded statements

Subroutines and Control Abstraction [2 Hrs]
Review of stack layout; Calling sequences; Parameter passing; Generic subroutines and modules; Exception handling; Coroutines; Events.

Imperative languages [5 Hrs]
Control Flow: **Expression evaluation; Structured and unstructured flow; Sequencing; Selection; Iteration; Recursion**; Non-determinacy, Structured Programming - Need and Design issues. Block Structures (Pascal), types arrays, records, sets, pointers, procedures, parameter passing, scope rules (in C).

UNIT-II

Object oriented languages[4 Hrs]

Grouping of data and Operations - Constructs for Programming Structures, abstraction Information Hiding, Program Design with Modules, Defined types, **Object oriented programming - concept of Object, inheritance, Derived classes and Information hiding - Templates- Exception handling (Using C++ and Java as example language).**

Functional Programming[8 Hrs]

Features, Implementation, Types - values and operations, Introduction to lambda calculus Product of types.Introduction to lambda calculus. Lists and Operations on Lists, Functions from a domain to a range, Function Application, Lexical Scope. Bindings of values and functions (Using Haskell/ Lisp as example language).Reactive programming and its concepts.

Logic Programming[2 Hrs]

Formal Logic Systems, Working with relations and their implementation (Using Prolog as example).
Database query Languages, Exception handling (Using SQL as example)

Concurrency[6 Hrs]

Background and motivation; Concurrency programming fundamentals; Implementing synchronization; Language-level mechanisms; Message passing. Run-Time Program Management: Virtual machines; late binding of machine code; Inspection/introspection.

Text Books:

1. Michael L. Scott: Programming Language Pragmatics, 3rd Edition, Elsevier, 2009.
2. Programming Language Design Concepts by David A. Watt, Wiley publications

Reference Books:

1. Ravi Sethi: Programming languages Concepts and Constructs, 2nd Edition, Pearson Education, 1996.
2. R Sebesta: Concepts of Programming Languages, 8th Edition, Pearson Education, 2008.
3. Allen Tucker, Robert Nonan: Programming Languages, Principles and Paradigms, 2nd Edition, Tata McGraw-Hill, 2007.
4. Programming Languages: Principles and Practice (English) 1st Edition (Paperback) by Kenneth C. Louden.

Department of Computer Science and Engineering

C422 **CS 1724 / CS 1736** **(2L +1T hrs/week)**

OPTIMIZATION TECHNIQUES

Questions to be set having equal weightage/marks from each unit: SIX
TWO from UNIT I, TWO from UNIT II, and TWO from both UNIT I and UNIT II combined.
Questions to be answered: FIVE
Compulsory TWO Questions from UNIT I and UNIT II, and ANY ONE from the combination Question

Objectives: The objective of this course is to understand the need and origin of the optimization methods and to get a broad picture of the various applications of optimization methods used in engineering. This course is indent for designing and controlling complex systems, solving hard problems of efficiently allocating scarce resources using in complete information, and developing sustainable strategies to master situations of conflict and co-operation using scientific methods and information technology.

Pre-requisites: Quantitative Analysis using C/C++, Design and Analysis of Algorithms and Probability& Statistics.

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

1. Demonstrate the knowledge and understanding of the basic ideas underlying optimization techniques.
2. Apply the mathematical approach to optimization problems relevant to engineering
3. Analyze the robustness of continuous linear optimization problems solutions using sensitivity analysis.
4. Ability to understand and interpret the results and information provided by a particular method.
5. Compare the robustness of continuous linear optimization problems solutions using various techniques

UNIT – I

Introduction to Operations Research [8 Hrs]
Introduction to OR modeling approach and various real life situations, Linear programming problems and applications, Solving Linear Programming problem using simultaneous equations and Graphical Method, Simplex Method and extensions, Sensitivity analysis - Duality theory. Transportation model, **Transshipment problems and Assignment problems.**

Dynamic Programming [6 Hrs]
Bellman's principle of optimality, Examples on the application on routing problem, Inventory problemSimplex problem, **Marketing problem.**

Network Analysis [6 Hrs]
PERT and CPM, Probability of achieving completion data, Cost analysis, Graph reduction theory, Updating, **Resource allocation. Resource smoothing.**

UNIT – II

Inventory Method [7 Hrs]
Inventory problem, Variables in an inventory problem, Inventory models with penalty, Storage andQuantity discount, **Safety stock**, Inventory models with probability, Demand, Multi item deterministic model. Simulation, Types of simulation models, **Applications of simulation for Inventory problems.**

Queuing Theory [7 Hrs]
Poison arrivals and Exponential service times, Waiting time and Idle time cost, Single channel and Multi-channel problem. **Applications of simulation for Queuing problems**. Monte Carlo technique applied to queuing problems, **Poisson arrivals and service time.**

Theory of Games [6 Hrs]

Introduction – Minimax (maximin) – Criterion and optimal strategy – Solution of games with saddle points– Rectangular games without saddle points – 2 X 2 games, Examples on the application of theory of games. – Dominance principle – mX2 & 2Xn games - Graphical method and Linear programming method for different problems. **Decision trees.**

Text Books:

1. Hamdy A. Taha, "Operations Research", Fifth edn. , Macmillan Publishing Company.
2. Kumar Gupta, Prem and Hira, D.S., "Operations Research", S Chand & Company Limited.
3. Swarup, Kanti, Gupta, P.K. and Manmohan, "Operations Research", Sultan Chand & Sons.

Reference Books:

1. Operations Research – Schaum outline series, MH
2. V.K. Kapoor-- Operations Research
3. Hiller F. and Leibermann G. J., "Operation Research", Holder Day Inc.
4. Srinath L.S., "PERT & CPM Principles and Applications", Affiliate East West Press (P).