Curriculum for Undergraduate Degree (B.Tech.) in Computer Science and Engineering (w.e.f. AY: 2020-21)

Part III: Detailed Curriculum

Fourth Semester

Course Name:	**Numerical Methods**		
Course Code:	BS-M 404	**Category:**	Basic Science Course
Semester:	4th	**Credit:**	2
L-T-P:	2-0-0	**Pre-Requisites:**	Some concepts from basic math – algebra, geometry, pre-calculus and statistics
Full Marks:	100		
Examination Scheme:	Semester Examination: 70	Continuous Assessment: 25	Attendance: 05

Course Objectives:	
1	To compute different numerical errors in computations.
2	To learn interpolation techniques.
3	To apply the techniques for solving integrations, ODEs.
4	Solve linear and non-linear equations.

Course Contents:

Module No.	Description of Topic	Contact Hrs.
1	**Approximation in numerical computation:** Truncation and rounding errors, Fixed and floating-point arithmetic, Propagation of errors.	2
2	**Interpolation:** Newton's Forward Interpolation, Newton's Backward Interpolation, Lagrange's Interpolation, Newton's Divided Difference Interpolation	4
3	**Numerical integration:** General Quadrature Formula, Trapezoidal Rule, Simpson's 1/3 Rule, Expression for corresponding error terms	3
4	**Numerical solution of a system of linear equations:** Gauss Elimination Method, Matrix Inversion, LU Factorization Method, Gauss-Seidel Iterative Method	6
5	**Numerical solution of Non-Linear equation:** Bisection Method, Regula-Falsi Method, Newton-Raphson Method	4
6	**Numerical solution of ordinary differential equation:** Euler's Method, Runge-Kutta Methods, Predictor-Corrector Methods, Finite Difference Method	5
7	**Measure of Central Tendency and Dispersion:** Mean, median, mode and S.D.	3
8	**Curve Fitting by Method of Least Square:** Fitting a straight line of the form $y = a + bx$, Fitting a curve of the form $y = ax + bx^2$, $y = ab^x, y = ae^{bx}, y = ax^b$.	3
Total		**30**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
2	Apply numerical methods to obtain approximate solutions to mathematical problems.
3	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
4	Analyse and evaluate the accuracy of common numerical methods.

Learning Resources:	
1	C.Xavier: C Language and Numerical Methods.
2	A. K. Jalan and Utpal Sarkar, Numerical Methods-A Programming Based Approach, Orient Blackswan Private Ltd.
3	Dutta & Jana: Introductory Numerical Analysis.
4	J.B.Scarborough: Numerical Mathematical Analysis.
5	Jain, Iyengar , & Jain: Numerical Methods (Problems and Solution).
6	Balagurusamy: Numerical Methods, Scitech.
7	Baburam: Numerical Methods, Pearson Education.
8	N. Dutta: Computer Programming & Numerical Analysis, Universities Press.
9	Soumen Guha & Rajesh Srivastava: Numerical Methods, OUP.
10	Srimanta Pal: Numerical Methods, OUP.

Course Name:	Computer Organization		
Course Code:	PC-CS401	Category:	Professional Core Course
Semester:	4th	Credit:	3
L-T-P:	3-0-0	Pre-Requisites:	Digital Electronics
Full Marks:	100		
Examination Scheme:	Semester Examination: 70	Continuous Assessment: 25	Attendance: 05

Course Objectives:	
1	To introduce students how Computer Systems work & basics involved in data representation.
2	This course will also expose students to the basic organization of Processor and Memory System.
3	The students will be able to know how I/O devices are being accessed.

Course Contents:		
Module No.	Description of Topic	Contact Hrs.
1	**Basic Computer Organization and Data Representation** Basic organization of the stored program computer and operation sequence for execution of a program. Role of operating systems and compiler/assembler. Fetch, decode and execute cycle, Concept of operator, operand, registers and storage.	5

	Commonly used number systems. Fixed- and floating-point representation of numbers Floating point - IEEE 754 standard, Overflow, Underflow	
2	**Microoperation and Computer Arithmetic:** Arithmetic Microoperations, Logic Microoperations, Shift Microoperation Design of adders - Ripple carry adder, Serial Adder and Carry Look Ahead Adder, BCD Adder Binary Incrementer Circuit, Binary Decrementer Circuit, Arithmetic Circuit, Arithmetic Logic Shift Unit Fixed point multiplication -Booth's algorithm. Fixed point division - Restoring and non-restoring algorithms.	9
3	**Central Processing Unit** General Register Organization, Stack Organization Instruction Formats, Addressing Modes, Instruction Set, CISC Characteristics, RISC Characteristics Design of control unit - hardwired and microprogrammed control.	6
4	**Memory Organization** Static and dynamic memory, Memory hierarchy, Associative memory. Cache memory, Associative Mapping, Direct Mapping, Set Associative Mapping, Virtual memory, Paging, Segmentation and Page replacement Algorithm, Memory unit design with special emphasis on implementation of CPU-memory interfacing. Data path design for read/write access.	10
5	**Input-Output Organization** Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Mode of Transfer, Priority Interrupt, Direct Memory Access	6
Total		**36L**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Describe Computer hardware System, Instruction sets and Addressing Mode.
2	Apply the knowledge of number system to perform different arithmetical operations.
3	Design memory organization that uses banks for different word size operations.
4	Compare different type of control units and I/O transfer techniques.

Learning Resources:	
1	Mano, M.M., "Computer System Architecture", PHI.
2	Hayes J. P., "Computer Architecture & Organisation", McGraw Hill,
3	Hamacher, "Computer Organisation", McGraw Hill,
4	William Stallings "Computer Organization and Architecture Designing for Performance", Pearson
5	B.Ram – "Computer Organization & Architecture", Newage Publications

Course Name:	**Operating System**		
Course Code:	PC-CS 402	**Category:**	Professional Core Course
Semester:	Fourth	**Credit:**	3
L-T-P:	3-0-0	**Pre-Requisites:**	Computer Organization
Full Marks:	100		
Examination Scheme:	Semester Examination: 70	Continuous Assessment: 25	Attendance: 05

Course Objectives:	
1	To Learn Operating System concepts and algorithms
2	To gain the knowledge about the application and analysis of algorithms

Course Contents:

Module No.	Description of Topic	Contact Hrs.
1	Introduction: Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.	3
2	Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads, Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR	6
3	Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, RAG, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	4
4	Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, The Producer Consumer Problem, Semaphores, Event Counters, Message Passing, Classical IPC Problems: Producer-Consumer Problem, Reader's & Writer's Problem, Dinning Philosophers Problem etc.	6
5	Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation– Fixed and variable partition– Internal and External fragmentation and Compaction; Paging, Protection and sharing, Disadvantages of paging, segmentation	6
6	Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Not recently used (NRU) and Least Recently used (LRU).	4
7	Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Boot-block, Bad blocks	4
8	File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table).	3
Total		**36L**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Recall and understand introductory concepts of operating system
2	Apply and analyze process scheduling methods and deadlock handling schemes
3	Understand inter process communication
4	Understand, apply and analyze memory management and disk management procedures

Learning Resources:	
1	Operating System Concepts, Silberschatz, Galvin and Gagne, Wiley
2	Principles of Operating System, Naresh Chauhan, Oxford
3	Operating System, Deitel, Deitel, Pearson

Course Name:	**Design and Analysis of Algorithm**		
Course Code:	PC-CS403	**Category:**	Mandatory Course
Semester:	Fourth	**Credit:**	3
L-T-P:	3-0-0	**Pre-Requisites:**	Data Structure, Discrete Mathematics, Basic Programming Ability
Full Marks:	100		
Examination Scheme:	Semester Examination: 70	Continuous Assessment: 25	Attendance: 05

Course Objectives:	
1	The aim of this course is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them
2	Through the complexity measures, different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood.

Course Contents:

Module No.	Description of Topic	Contact Hrs.
1	**Introduction:** Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Method of Iteration, Recursion Tree method and Masters' theorem (Examples: Analysis of Binary Search, Merge Sort and Quick Sort using Recurrence)	6
2	**Fundamental Algorithmic Strategies:** **Divide and Conquer Method:** Basic method, use, Example – Max-Min Problems and its complexity analysis. **Greedy Method:** Basic method, use, Examples – Fractional Knapsack Problem, Job sequencing with deadlines, Activity Scheduling Problem, Travelling Salesperson Problem and their complexity analysis **Dynamic Programming:** Basic method, use, Examples – Matrix Chain Manipulation, 0/1 Knapsack Problem and their complexity	10

	analysis **Branch and Bound and Backtracking:** Basic method, use, Examples – 15 Puzzles Problem, N queens problem, Graph Coloring problem, Hamiltonian Cycle Problem	
3	**Graph and Tree Algorithms:** Traversal algorithms: Recapitulation of Depth First Search (DFS) and Breadth First Search (BFS); Shortest path Algorithms (Single Source and All Pairs with their Complexity Analysis), Transitive Closure, Minimum Spanning Tree (Prim's and Kruskal's Algorithms with their Complexity Analysis), Topological Sorting, Ford Fulkerson algorithm, Max-Flow Min-Cut theorem (Statement and Illustration).	10
4	**Tractable and Intractable Problems:** Computability of Algorithms, Computability classes – P, NP, NP- complete and NP-hard. Satisfiability Problem, Cook's theorem, Clique decision problem	6
5	**Advanced Topics:** Approximation Algorithms: Introduction and Example - Vertex Cover Problem, Randomized Algorithms: Introduction and Example - Quick Sort	4
Total		**36**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Recall the fundamental concepts of Asymptotic Notations and identify their mathematical significance and analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms. Derive and solve recurrence relation.
2	Describe different algorithm design techniques like D&C, Greedy Method, DP, Backtracking, Branch and Bound, Graph Algorithms, NP etc and their implementations.
3	Apply appropriate algorithms and required Data Structure to construct the solution of a given problem.
4	Explain Randomized algorithms (expected running time, probability of error), and Approximation algorithm to compute approximation factors.
5	Analyze algorithms and determine the correctness.

Learning Resources:	
1	Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2	Fundamentals of Algorithms – E. Horowitz et al.
3	Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
4	Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
5	Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA
6	Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House (AICTE Recommended Textbook – 2018)
7	Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai

Course Name:	**Discrete Mathematics**		
Course Code:	PC-CS 404	**Category:**	Basic Science Course
Semester:	4th	**Credit:**	3
L-T-P:	3-0-0	**Pre-Requisites:**	Some concepts from basic math – algebra, geometry, pre-calculus
Full Marks:	100		
Examination Scheme:	Semester Examination: 70	Continuous Assessment: 25	Attendance: 05

Course Objectives:	
1	To use mathematical logics and Boolean algebra in the field of computer applications.
2	To know about Set-Relation-Function and Group theory.
3	To learn counting techniques and number theory.
4	To use the concept of graph theory in engineering problems.

Course Contents:

Module No.	**Description of Topic**	**Contact Hrs.**
1	Module-1: **Sets-Relation-Function** • Operations and Laws of Sets • Cartesian Products, Binary Relation, Equivalence Relation, Partial Ordering Relation, Lattice **Number Theory** • Proofs by Mathematical Induction • The Division Algorithm, Prime Numbers, The Greatest Common Divisor, Euclidean Algorithm, The Fundamental Theorem of Arithmetic	10
2	**Module-2:** **Combinatorics** • Basic Counting Techniques, Inclusion and Exclusion Theorem • Permutation and Combination • Pigeon-Hole Principle	6
3	**Module-3:** **Propositional Logic and Proofs** • Basic Connectives and Truth Tables of propositional logics, Disjuntive and Conjunctive Normal Form using truth table, Argument • Quantifiers and their uses • Proofs; Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof by Mathematical Induction • The Laws of Logic, Logical Implication, Rules of Inference	8
4	**Module-4:** **Algebraic Structures and Boolean Algebra** • Algebraic Structures with one Binary Operator • Group, Subgroup, Cyclic group, Permutation group, Symmetric group. • Coset, Lagrange's Theorem, Normal Subgroup, Quotient group • Homomorphism and Isomorphism of groups **Algebraic Structures with two Binary Operators**	10

	• Rings, Integral Domain and Fields **Boolean Algebra** • Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjuntive and Conjuntive Normal Form • Switching network from Boolean expression using Logic Gates • Karnaugh Map	
5	**Module-5:** **Advanced Graph Theory** • Planar and Dual graph: Kuratowski's graphs, Euler's formulae for connected and disconnected planar graphs, Detection of planarity • Graph Coloring: Vertex coloring, Chromatic number of complete graphs, circuit and bipartite graph, Chromatic polynomial • Connectivity and matching	6
Total		**40**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Express a logic sentence in terms of predicates, quantifiers, and logical connectives
2	Derive the solution for a given problem using deductive logic and prove the solution based on logical inference
3	Classify its algebraic structure for a given a mathematical problem,
4	Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5	Develop the given problem as graph networks and solve with techniques of graph theory

Learning Resources:	
1	Russell Merris, Combinatorics, Wiley-Interscience series in Discrete Mathematics and Optimisation
2	N. Chandrasekaran and M. Umaparvathi, Discrete Mathematics, PHI
3	Gary Haggard, John Schlipf and Sue Whitesides, Discrete Mathematics for Computer Science, CENGAGE Learning
4	Gary Chartrand and Ping Zhang – Introduction to Graph Theory, TMH
5	J.K. Sharma, Discrete Mathematics, Macmillan
6	Winfried Karl Grassmann and Jean-Paul Tremblay, Logic and Discrete Mathematics, PRSEAON.
7	S. K. Chakraborty and B. K. Sarkar, Discrete Mathematics, OXFORD University Press.
8	Douglas B. West, Introduction to graph Theory, PHI
9	C. L. Liu, Elements of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill, 2000.
10	R. C. Penner, Discrete Mathematics: Proof Techniques and Mathematical Structures, World Scientific, 1999.
11	R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, 2nd Ed., Addison-Wesley, 1994.
12	N. Deo, Graph Theory, Prentice Hall of India, 1974.
13	S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill, 1999.
14	J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill, 1997.
15	Higher Algebra- S.K. Mapa
16	N. Chandrasekaran and M. Umaparvathi, Discrete Mathematics, PHI
17	S.B. Singh, Discrete Structures – Khanna Publishing House (AICTE Recommended Textbook – 2018)

18	S.B. Singh, Combinatorics and Graph Theory, Khanna Publishing House (AICTE Recommended Textbook – 2018)

Course Name:	**Numerical Methods Lab**		
Course Code:	BS-M 494	**Category:**	Basic Science Course
Semester:	4th	**Credit:**	1
L-T-P:	0-0-2	**Pre-Requisites:**	
Full Marks:	100		
Examination Scheme:	Semester Examination: 60	Continuous Assessment: 35	Attendance: 05

Course Objectives:	
1	To compute different numerical errors in computations.
2	To learn interpolation techniques.
3	To apply the techniques for solving integrations, ODEs.
4	Solve linear and non-linear equations.

Course Contents:

Module No.	Description of Topic	Contact Hrs.
1	**Assignments on Interpolation:** Newton's Forward Interpolation, Newton's Backward Interpolation, Lagrange's Interpolation	6
2	**Assignments on Numerical Integration:** Trapezoidal Rule, Simpson's 1/3 Rule	6
3	**Assignments on Solution of Transcendental Equations:** Bisection Method, Regula-Falsi Method, Newton-Raphson Method	6
4	**Assignments on ODEs:** Euler's Method, Runge-Kutta Method of Order Four	6
5	**Curve Fitting by the Method of Least Squares:** Fitting a straight line of the form $y = a + bx$, Fitting a curve of the form $y = ax + bx^2$, $y = ab^x, y = ae^{bx}, y = ax^b$	6
6	**Measure of Central Tendency:** Mean and Standard Deviation, Median and Mode	3
7	**Assignments on Numerical Solution of a system of Linear Equations:** Gauss Elimination Method, Gauss-Seidel Method	3
Total		**36**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
2	Apply numerical methods to obtain approximate solutions to mathematical problems.
3	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
4	Analyse and evaluate the accuracy of common numerical methods.

Learning Resources:	
1	C. Xavier: C Language and Numerical Methods.
2	A. K. Jalan and Utpal Sarkar, Numerical Methods-A Programming Based Approach, Orient Blackswan Private Ltd.
3	Dutta & Jana: Introductory Numerical Analysis.
4	J.B.Scarborough: Numerical Mathematical Analysis.
5	Jain, Iyengar , & Jain: Numerical Methods (Problems and Solution).
6	Balagurusamy: Numerical Methods, Scitech.
7	Baburam: Numerical Methods, Pearson Education.
8	N. Dutta: Computer Programming & Numerical Analysis, Universities Press.
9	Soumen Guha & Rajesh Srivastava: Numerical Methods, OUP.
10	Srimanta Pal: Numerical Methods, OUP.

Course Name:	**Computer Organization Lab**		
Course Code:	PC-CS491	**Category:**	Professional Core Course
Semester:	3rd	**Credit:**	1.5
L-T-P:	0-0-3	**Pre-Requisites:**	Digital Electronics
Full Marks:	100		
Examination Scheme:	Semester Examination: 60	Continuous Assessment: 35	Attendance: 05

Course Objectives:	
1	To Familiar with different type of IC-chips
2	To Design different arithmetic and Logic Circuits
3	To know the working principle of RAM IC.

Course Contents:		
Module No.	**Description of Topic**	**Contact Hrs.**
1	Familiarity with IC-chips: a) Multiplexer, b) Decoder, c) Encoder d) Comparator Truth Table verification and clarification from Data-book.	6
2	Design an Adder/Subtractor composite unit.	3
3	Design a BCD adder.	3
4	Design of a 'Carry-Look-Ahead' Adder circuit.	3
5	Design of Code Converter Circuit	3
6	Construct a Binary Multiplier using basic logic gates	3
7	Use a multiplexer unit to design a composite ALU	3
8	Use ALU chip for multibit arithmetic operation	3
9	Implement read write operation using RAM IC	3
10	(a) & (b) Cascade two RAM ICs for vertical and horizontal expansion.	6
Total		**36**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Verify truth-table of different types of IC.
2	Design different type of adder circuits.
3	Design ALU by applying the knowledge of Combinational circuit.
4	Design different circuits with RAM ICs and perform read-write operation.

Learning Resources:	
1	Morris Mano- Digital Logic and Computer Design- PHI
2	Leach & Malvino—Digital Principles & Application, 5/e, McGraw H
3	S. Salivahanan, S. Arivazhagan – Digital Circuits and Design
4	Mano, M.M., "Computer System Architecture", PHI.

Course Name:	**Operating System Lab**		
Course Code:	PC-CS 492	**Category:**	PC
Semester:	4th	**Credit:**	1.5
L-T-P:	0-0-3	**Pre-Requisites:**	Computer organization
Full Marks:	100		
Examination Scheme:	Semester Examination: 60	Continuous Assessment: 35	Attendance: 05

Course Objectives:	
1	To learn UNIX commands and shell script
2	To gain the knowledge about process, thread, signal, semaphore and IPC

Course Contents:		
Module No.	**Description of Topic**	**Contact Hrs.**
1	UNIX Commands and Permissions	3
2	Creating a bash shell script, making a script executable, shell syntax (variables, conditions, control structures, functions, CLA, String)	9
3	C programs for parent process, child process, orphan process, sleeping process, running process, zombie process.	6
4	Multithreaded C program using PThread API and Win32 API	6
5	C programs for signal handling, sending signals and signal interface.	3
6	C programs regarding Semaphore	3
7	Inter-process communication through shared memory segment, message queues, pipes and named pipes	6
Total		**36P**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Recall and understand UNIX commands and applications of shell script
2	Apply and Analyze Process and Thread execution
3	Apply and Analyze Signal and Semaphore
4	Apply and Analyze IPC related concepts

Learning Resources:	
1	UNIX Concepts and Applications, Sumitabha Das, McGrawhill
2	Vijay Mukhi's The C Odyssey UNIX – The Open Boundless C, BPB Publications

Course Name:	**Design and Analysis of Algorithm lab**		
Course Code:	PC-CS493	**Category:**	Professional Core Courses
Semester:	Fourth	**Credit:**	1.5
L-T-P:	0-0-3	**Pre-Requisites:**	Data Structure, Basic Programming Ability
Full Marks:	100		
Examination Scheme:	Semester Examination: 60	Continuous Assessment: 35	Attendance: 05

Course Objectives:	
1	The aim of this course is to study about various designing paradigms of algorithms for solving real world problems.
2	Through this course one can apply appropriate algorithms and methods of analysis.
3	To pick an appropriate data structure for a design situation is also under consideration.

Course Contents:		
Module No.	**Description of Topic/ Experiment**	**Contact Hrs.**
The contents should include about 10 assignments with the focus given as outlined below:		
UNIT - I Divide and Conquer, Greedy Method, Dynamic Programming		
1	Implement Binary Search, Merge Sort, Implement Quick Sort, Find Maximum and Minimum Element from an Array of Elements Implement Knapsack Problem, Job sequencing with deadlines, Traveling Salesman Problem Find the minimum number of scalar multiplication needed for Chain of Matrix	15
UNIT - II Graph Traversal Algorithm, Minimum Cost Spanning Tree Generation Algorithms, Shortest Path Algorithms		
2	Implement Breadth First Search (BFS), Depth First Search (DFS) Implement Minimum Cost Spanning Tree by Prim's and Kruskal's Algorithm Implement Single Source shortest Path for a graph (Dijkstra, Bellman Ford Algorithm) and All pair of Shortest path for a graph (Floyd- Warshall Algorithm)	15
UNIT - III Backtracking and Branch and Bound		

3	Implement N Queen problem Implement Graph Coloring Problem Implement Hamiltonian Problem Implement 15-Puzzle Problem	6
Total		**36**

Course Outcomes:	
After completion of the course, students will be able to:	
1	Demonstrate and implement Binary Search, Merge Sort, Quick Sort, and Max-min Problem using D&C Algorithm Design Techniques.
2	Implement Fractional Knapsack, Job Sequencing with Deadline, TSP, Matrix Chain, Graph Traversals, MST problems, Shortest Path, N- Queens, Graph Coloring, Hamiltonian Cycle, and 15 Puzzles using proper Algorithm Design Techniques.
3	Apply suitable algorithm for solving a particular problem.
4	Analyze the complexities and memory usages of different algorithms.

Learning Resources:	
1	Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2	Fundamentals of Algorithms – E. Horowitz et al.
3	Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai
4	Design and Analysis of Algorithm, Biswas and Dey, JBBL

Course Name:	**Constitution of India**		
Course Code:	MC472	**Category:**	Mandatory Course
Semester:	Fourth	**Credit:**	Zero
L-T-P:	2-0-0	**Pre-Requisites:**	
Full Marks:	100		
Examination Scheme:	Semester Examination of 100 marks		

Course Objectives:	
1	Develop an understanding of the nation's constitution.
2	Develop knowledge about the various levels of governance in the country.

Course Contents:		
Module No.	**Description of Topic**	**Contact Hrs.**
1	**Introduction:** : Sources and Constitutional history. Preamble, Fundamental Rights and Duties, Directive Principles of State Policy.	3
2	**Union Government and its Administration :** Structure of the Indian Union: Federalism, Centre- State relationship, President: Role, power and position, PM and Council of ministers, Lok Sabha, Rajya Sabha Supreme Court	6
3	**State Government and its Administration**	6

	Governor.Role and Position, CM and Council of ministers High Court	
4	**Local Administration District's Administration head:** Role and Importance, Municipalities: Introduction, Mayor, and role of Elected Representative. Pachayati raj: Introduction, Zila Pachayat, Elected officials and their roles. Importance of grass root democracy	6
5	**Election Commission Election Commission:** Role and Functioning, Chief Election Commissioner	2
Total		

Course Outcomes:	
After completion of the course, students will be able to:	
1	Gain an understanding of the constitution of India.
2	Become aware of the various levels of governance in the country.

Learning Resources:	
1	'Indian Polity' by Laxmikanth
2	'Indian Administration' by Subhash Kashyap
3	'Indian Constitution' by D.D. Basu
4	'Indian Administration' by Avasti and Avasti