Himachal Pradesh University, Shimla
(‘A’ Grade, NAAC Accredited)

Scheme of Examination and Syllabus of
MASTER OF TECHNOLOGY in COMPUTER SCIENCE
M. Tech. (Computer Science)
(CBCS)

DEPARTMENT OF COMPUTER SCIENCE

CBCS CURRICULUM (2025-26)

MASTER OF TECHNOLOGY in COMPUTER SCIENCE

M. Tech. (Computer Science)(CBCS)

(For the Batches Admitted 2025 Onwards)

DEPARTMENT OF COMPUTER SCIENCE
HIMACHAL PRADESH UNIVERSITY, SHIMLA

VISION

Pursue conducive advancement towards nurturing globally competent and ethically conscientious professionals and entrepreneurs in agile computing technologies and allied spheres for unceasing evolution of Nations IT affiliated commercial and research endeavours.

MISSION

Thrive to establish a strong foundation for technical competency in spheres concordant to software oriented design and development. Nurture skills and competency for administering expertise gained in computing discipline to a wide horizon of interdisciplinary application domains, thus supporting sustainable development of the society. Habituate the students to strive for technological innovations and successful endeavours ethically, supported by sustained learning continuance and problem solving proficiency that may promote nations welfare in terms of economic acceleration leading to the growth of society.

NAME OF THE PROGRAMME: MASTER OF TECHNOLOGY in COMPUTER SCIENCE
M. Tech. (Computer Science)
DURATION : TWO YEARS

PROGRAMME OUTCOMES (POs)	
PO1	Capable of demonstrating comprehensive disciplinary knowledge gained during course of study.
PO2	Capability to ask relevant/appropriate questions for identifying, formulating and analyzing the research problems and to draw conclusion from the analysis.
PO3	Ability to communicate effectively on general and Technical topics with the engineering community and with society at large.
PO4	Capability of applying knowledge to solve Engineering and other problems.
PO5	Capable to learn and work effectively as an individual, and as a member or leader in diverse teams, in multidisciplinary settings.
PO6	Ability of critical thinking, analytical reasoning and research based knowledge including design of experiments, analysis and interpretation of data to provide conclusions.
PO7	Ability to use and learn techniques, skills and modern tools for scientific and engineering practices.
PO8	Ability to apply reasoning to assess the different issues related to society and the consequent responsibilities relevant to the professional Engineering practices.
PO9	Aptitude to apply knowledge and skills that are necessary for participating in learning activities throughout life.
PO10	Capability to identify and apply ethical issues related to one’s work, avoid unethical behaviour such as fabrication of data, committing plagiarism and unbiased truthful actions in all aspects of work.
PO11	Ability to demonstrate knowledge and understanding of the engineering principles and apply these to manage projects.

PROGRAMME SPECIFIC OUTCOMES (PSOs)	
PSO1	Supplement potential for pursuing advanced studies, engaging in research & technological development directed towards innovative activities, and nurturing entrepreneurial skills.
PSO2	Strengthen competency for innovating solutions to real-world problems by exercising data analysis skills and adopting contemporary technologies for demanding prospective applications.
PSO3	Inculcate the practice to administer Professional & Ethical virtues, along with Social and Environmental regulations.
PSO4	Stimulate the aptitude for problem analysis and programming skills for computer based system design and modeling in allied spheres related to Algorithmic, Computational, Architectural and Database environments, along with emerging technologies such as Machine Learning& Intelligent systems , Evolutionary Techniques and Optimization, Data Science & Analytics, Distributed and Wireless Communication in cognation with IoT and Cloud Computing , Web and Mobile application designing, and Real World Enhancement using Computer Vision & Augmented Reality.

Duration: 2 Years (4 Semesters)

About the Programme:

The M. Tech. (Computer Science) is a two years full time post-graduation course spread over four semesters. In this programme, the students will be taught, the core and advanced subjects of Computer Science.

Eligibility:

Candidates, who have passed Master degree with minimum of 55% marks (50% marks for SC/ST category), or equivalent Grade point, in Mathematics/ Physics/ Electronics/ Computer Applications (MCA)/ Computer Science/ Information Technology/ 4-years Bachelor's degree in Engineering/ Technology.

Or

Any examination, of university in foreign country, recognized as equivalent for the above purpose by equivalence committee of its own or on recommendation of Association of Indian Universities with 55% marks (50% marks for SC/ST).

Mode of Selection:

The admission to M. Tech. (Computer Science) will be through the merit of an entrance test of duration one and half hours to be conducted by H.P. University, Shimla. The entrance test shall be of 100 MCQ type questions of one mark each, comprising of the following components:

Contents	Marks	Duration
Computer Science	100	90 Minutes

The Candidates who have qualified GATE/ NET/ SLET in Computer Science or Engineering will be given 20% extra of the marks obtained in the entrance test for calculating the merit for the purpose of admission.

The minimum qualifying marks in the Entrance Examination (written test) shall be 40% i.e. 40 marks, (35% i.e. 35 marks for SC and ST) out of total of 100 marks.

No. of Seats:

Seats	: 18 + 05 (In-service)*
Supernumerary Seats	: As per University Rules.

Reservation for M.Tech.:

106 Points Roster will be followed as per the HPU Hand Book of Information (HBI).

In-service Candidates:

** 5 seats are reserved for in-service applicants.*

The 'in-service' candidate shall be a candidate who is an employee (on a regular or contract basis) of Govt. of Himachal Pradesh (including Colleges & Universities under the Govt.) or Govt. of India or Boards/Corporations under state or centre governments having their headquarters in the state and is serving the organization in which currently employed for at least last 2 years. An in-service candidate has to submit the following along with the application form:

(a) The appointment letter from the employer and a certificate from the employer that he/she has been working with the concerned organization for at least last two years and shall continue to be on their roll for at least whole of the duration of the course,
(b) NOC on the prescribed Performa from his/her employer that in case the candidate is selected for the admission, he/she shall be granted a Study leave for a period of two years from the date of the commencement of the course as provided by the department. Upon confirmation of the admission, a

letter on the prescribed Performa (provided by the department) from the employer, sanctioning study leave for a period of 2 years is to be submitted in the department on or before the stipulated last date of admission along with an affidavit from the court duly signed by the notary clearly stating that if the information supplied in any of the above mentioned documents is found to be contrary to the facts at any stage, the candidate shall be held responsible for it and the university shall have the right to cancel the admission/degree of the candidate.

The admission of such candidates shall be confirmed only if a letter from the employer sanctioning the study leave of two years is deposited within the stipulated period for the admission as mentioned in the prospectus. The validity of the above mentioned documents shall be assessed by the department and the department shall also have the right of corresponding with the employer of the candidate in case of any doubts regarding these documents. The decision of the department in rejecting or accepting these documents shall be final. In any case, notwithstanding the admission to the said course, if the department finds that the candidate has supplied false or misleading information through any of the above mentioned documents, the university shall have the right to a) cancel the admission/degree of the candidate, and b) to initiate civil/criminal proceeding against the candidate.

Scheme of Examination:

English will be the medium of instruction and examination.

(i) Each theory paper shall be of 3 hours duration and shall carry 100 marks (75 marks for end semester theory examination and 25 marks for continual internal assessment).
(ii) In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.
(iii) Each practical paper shall be of 3 hours duration and will carry 100 marks (75 marks for end semester practical examination and 25 marks for continual internal assessment).
(iv) In the 3rd semester each student shall be:

(a) Attached to a teacher of the Department who shall act as a supervisor for seminar, Pre- Dissertation and Dissertation work of the students in 3rd& 4th semesters.
(b) Assigned a topic by the guide and the students are required to prepare and present a seminar.
(c) Assigned a topic for carrying out research which normally may be related to the findings of a thorough study on a research area of significance which is relevant to the field of Computer Science. At the end of 3rd semester, each student has to submit (3+1) nos. of copies of spiral bound report of his findings on the topic allotted to him/her in the office of the Department along with the softcopy of the same on a CDROM/DVD.
(d) Assessed and evaluated for Pre- Dissertation by an external examiner.

(v) In the 4th semester, each student is required to:

(a) Work on a topic in a significant area of research relevant to the field of Computer Science for dissertation under the guidance of his/her guide.
(b) Submit 4 nos. of hard bound copies of the dissertation in the office of the Department along with the softcopy of the same on a CDROM/DVD.
(c) To ensure the quality of the research work carried out by them in the 3rd& 4th semester, each student, hence is required to participate in some relevant regional/national level conferences/international level/ workshops/ seminars during these semesters. Therefore, it is mandatory for each student to publish at least one research paper (Full paper or abstract or extended abstract in the proceedings in print version/ e-version) in a regional/ national level journal/ conference/ International Journal/ workshop/ Seminar. The certificate/ proof of the published paper along with the paper shall form the part of the submitted dissertation. The students shall not be allowed to submit their dissertation without completing this requirement.

(vi) The dissertation of each candidate shall be sent to the external examiner well in advance by the Department of Computer Science, H.P. University before the conduct of the viva-voce examination.
(vii) The dissertation will be jointly evaluated by internal guide and external examiner.
(viii) In respect of theory papers 25 marks in each paper shall be reserved for award of internal assessment based on evaluation procedure for internal assessment marks as Two Mid Term Test should be conducted by the concerned teacher each of 10 marks. Five marks may be given by the concerned

teacher on the basis of performance during the course (Seminar/ assignments/ interactions/ attendance etc.).
(ix) Attendance: In order to be eligible to appear in the university examinations a candidate should have secured 75% attendance in each of the concerned subjects (theory as well as practical).

The credits for the first year are 52 (26+26) and for the second year are 52 (26+26). Total credits of the course will be 52+52 = 104.

HIMACHAL PRADESH UNIVERSITY, SHIMLA
SCHEME OF EXAMINATIONS FOR
MASTER OF TECHNOLOGY in COMPUTER SCIENCE
M. Tech. (Computer Science)
CHOICE BASED CREDIT SYSTEM (CBCS)
W. E. F. ACADEMIC SESSION 2025-26

First Semester

Paper No.	Title	Credits	Periods per week	Exam Time (Hrs.)	Max Marks (Theory)	Continual Internal Assessment	Total Marks
MT-CS-25-11	Computer Architecture and Parallel Processing	4	4	3	75	25	100
MT-CS-25-12	Software Engineering	4	4	3	75	25	100
MT-CS-25-13	Computer Oriented Optimization Method	4	4	3	75	25	100
MT-CS-25-14	Data Structure & Algorithm Analysis	4	4	3	75	25	100
	Elective-I	4	4	3	75	25	100
MT-CS-25-16	Practical on MT-CS-25-13	3	6	3	75	25	100
MT-CS-25-17	Practical on MT-CS-25-14	3	6	3	75	25	100
Total		**26**					**700**
Elective-I							
MT-CS-25-15(i)	Operating system and Case Study	4	4	3	75	25	100
MT-CS-25-15(ii)	Advanced Software Engineering Concepts	4	4	3	75	25	100
MT-CS-25-15(iii)	Cyber Law	4	4	3	75	25	100

Second Semester

Paper No.	Title	Credits	Periods per week	Exam Time (Hrs.)	Max Marks (Theory)	Continual Internal Assessment	Total Marks
MT-CS-25-21	Research Methodology in Computer Science	4	4	3	75	25	100
MT-CS-25-22	Research Ethics	4	4	3	75	25	100
MT-CS-25-23	Distributed Data Base Management System	4	4	3	75	25	100
MT-CS-25-24	Machine Learning using Python	4	4	3	75	25	100
	Elective-II	4	4	3	75	25	100
MT-CS-25-26	Practical on MT-CS-25-23	3	6	3	75	25	100
MT-CS-25-27	Practical on MT-CS-25-24	3	6	3	75	25	100
Total		**26**					**700**
Elective-II							
MT-CS-25-25(i)	Artificial Intelligence & Expert System	4	4	3	75	25	100
MT-CS-25-25(ii)	Big Data Analytics	4	4	3	75	25	100
MT-CS-25-25(iii)	Cloud Computing	4	4	3	75	25	100

Third Semester

Paper No.	Title	Credits	Periods per week	Exam Time (Hrs.)	Max Marks (Theory)	Continual Internal Assessment	Total Marks
MT-CS-25-31	Computer Networks	4	4	3	75	25	100
MT-CS-25-32	Data Warehousing and Data Mining	4	4	3	75	25	100
	Elective-III	4	4	3	75	25	100
					External Evaluation	**Internal Evaluation**	
MT-CS-25-34	Pre - Dissertation	14	-	-	-	-	-
(a)	Seminar	-	-	-	-	50	50
(b)	Documentation	-	-	-	50	-	50
(c)	Presentation & Viva	-	-	-	200	-	200
Total		**26**					**600**
		Elective-III					
MT-CS-25-33(i)	Software Quality and Testing	4	4	3	75	25	100
MT-CS-25-33(ii)	Automata Theory and Compiler Design	4	4	3	75	25	100
MT-CS-25-33(iii)	Graph Theory	4	4	3	75	25	100

Fourth Semester

Paper No.	Title	Credits	Periods per week	Exam Time (Hrs.)	External Evaluation	Internal Evaluation	Total Marks
MT-CS-25-41	Dissertation	26	-	-	-	-	-
(a)	Seminar (Two)	-	-	-	-	100	100
(b)	Documentation	-	-	-	100	-	100
(c)	Presentation & Viva	-	-	-	400	-	400
Total		**26**					**600**

MT-CS-25-11: Computer Architecture and Parallel Processing L:4 T:0 P:0

Course Objectives: This course explores the essential principles of computer architecture, covering topics such as register transfer and micro-operations, the fundamental design of computers, micro-programmed control, CPU architecture, parallel processing, memory management, input/output systems, pipelining concepts, and multiprocessor design and programming. Through this course, students will develop a deep understanding of how modern computer systems are structured, organized, and function, allowing them to grasp the core principles governing computational tasks and parallel processing methods.

UNIT-I

Register Transfer And Micro Operations: Register Transfer Language, Bus and Memory Transfer-Three-State Bus Buffers, Memory Transfer, Arithmetic Microoperations-Binary Adder, Binary Adder-Subtractor, Binary Incrementer, Arithmetic Circuit, Logic Microoperations, Shift Microoperations, Arithmetic Logic Shift Unit.

Basic Computer Organization and Design: Instruction Codes, Computer Registers and Instructions, Timing and Control, Instruction Cycle, Memory Reference Instructions, Input–Output and Interrupts, Design of Basic Computer, Design of Accumulator Logic.

UNIT-II

Microprogrammed Control: Control memory, Address sequencing, Microprogram example- Computer configuration, Microinstruction format, Symbolic microinstruction, The Fetch Routine, Symbolic microprogram, Binary microprogram, Design of control unit.

Central Processing Unit: Introduction to CPU, General Register Organization, Stack Organization-Register Stack, Memory Stack, Reverse Polish Notation, Evaluation of Arithmetic Expressions, Instruction formats- Three Address Instruction, Two Address Instruction, One Address Instruction, Zero Address Instruction, Risk Instruction, Addressing modes, Data Transfer and Manipulation – Data Transfer Instructions, Data Manipulation Instructions, Arithmetic Instructions, Logical and Bit Manipulation Instructions, Shift Instructions. Program Control, Reduced Instruction Set Computer-RISC, CISC.

UNIT-III

Introduction to Parallel Processing: Evolution of Computer Systems, Parallelism in Uniprocessor Systems, Parallel Computer Structures, Architectural Classification Schemes, Parallel Processing Applications.

Memory and Input Output Subsystems: Hierarchical Memory Structure, Virtual Memory System, Memory Allocation and Management, Cache Memory and Management, Input Output Subsystems.

UNIT-IV

Principles of Pipelining and Vector Processing: Pipelining: An Overlapped Parallelism, Instructions and Arithmetic Pipelines, Principles of Designing Pipelined Processors. Vector Processing Requirements.

Multiprocessor Architecture and Programming: Functional Structures- Loosely and Tightly Coupled Multiprocessors, Processor Characteristics for Multiprocessing, Interconnection Networks, Parallel Memory Organizations, Multi Processor Operating Systems, Exploiting Concurrency for MultiProcessing.

Text Book:

1. M. Morris Mano, "Computer System Architecture", Pearson Education, 2004.
2. Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", McGraw Hill, 1993.

Reference Book:

1. Kai Hwang and Faye A. Briggs, "Computer Architecture and Parallel Processing. McGraw Hill, 1985.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understanding of Computer Architecture Fundamentals.
CO2: Proficiency in Designing Basic Computer Components.
CO3: Knowledge of Advanced CPU Concepts.
CO4: Understanding of Parallel Processing Principles.
CO5: Proficiency in Memory Systems and Input-Output Subsystems.
CO6: Ability to Analyze and Design Computer Systems.
CO7: Preparation for Advanced Studies and Research.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-12: Software Engineering **L:4 T:0 P:0**

Course Objectives: This course explores the evolving landscape of software engineering, focusing on the changing nature of software, software development processes, requirements analysis, software architecture, design methodologies, coding practices, software metrics, and software maintenance. Students will gain a comprehensive understanding of the principles, techniques, and tools essential for designing, building, and maintaining high-quality software systems.

UNIT-I

Evolving Role of Software, Software Engineering, Changing nature of Software, Software Myths, Terminologies, Role of management in software development Software Process and desired Characteristics, Software Life Cycle Models: Build & Fix Model, Water Fall Model, Incremental Process Model, Evolutionary Process Models, Unified Process, Comparison of Models, Other Software Processes, Selection of a Model.

Software Requirements Analysis & Specifications: Requirements Engineering, Types of Requirements, Feasibility Studies, Requirements Elicitation, Requirements - Analysis Documentation, Validation and Management.

UNIT-II

Software Architecture: Its Role, Views, Component & Connector View and its architecture style, Architecture Vs Design, Deployment View & Performance Analysis, Documentation, Evaluation Software Project Planning: Size estimation, Cost Estimation, COCOMO, COCOMO – II, Software Risk Management.

UNIT-III

Function Oriented Design: Design principles, Module level Concepts, Notation & Specification, Structured Design Methodology, Verification.

Object-Oriented Design: OOAnalysis & Design, OO Concepts, Design Concepts, UML – Class Diagram, Sequence & Collaboration Diagram, Other diagrams & Capabilities, Design Methodology – Dynamic and Functional Modeling, Internal Classes & Operations.

Detailed Design: PDL, Logic/Algorithm Design, State Modeling of Classes, Verification –Design Walkthroughs, Critical Design Review, Consistency Checkers.

UNIT-IV

Coding: Programming Principles & Guidelines, Coding Process, Refactoring, Verification.

Software Metrics: What & Why, Token Count, Data Structure Metrics, Information Flow Metrics, Object-Oriented Metrics, Use Case Oriented Metrics, Web Engineering Project Metrics, Metric Analysis.

Software Maintenance & Certification: Maintenance, Maintenance Process and Models, Estimation of Maintenance Costs, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation, Requirements of Certification, Types.

Text Books:

1. Pankaj Jalote, “An Integrated Approach to Software Engineering”, 3rd Edition, Narosa Publishing House, 2005.
2. K.K. Aggrawal and Yogesh Singh, “Software Engineering”, 3rd Edition, New Age International (P) Ltd, 2008.

Reference Books:

1. Pressman, R.S., “Software Engineering – A Practitioner's Approach”, Third Edition, McGraw Hills, 2008.

2. Mall Rajib, “Fundamentals of Software Engineering”, PHI, New Delhi, 2005.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understanding of Software Engineering Fundamentals.
CO2: Proficiency in Designing Computer Software.
CO3: Knowledge of Advanced Software Engineering Concepts.
CO4: Understanding of Software Project Planning.
CO5: Proficiency in Programming Principles of Software.
CO6: Ability to Analyze Software Project Risks.
CO7: Preparation for Advanced Studies and Research.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-13: Computer Oriented Optimization Method L:4 T:0 P:0

Course Objectives: This course introduces students to the principles and techniques of Operations Research and management, focusing on optimization methods, linear programming, simplex method, dual simplex method, revised simplex method, networking scheduling using PERT/CPM, transportation problem, game theory, and inventory control. Students will learn how to apply mathematical models and algorithms to solve complex decision-making problems commonly encountered in various domains.

UNIT-I

Introduction to O.R. – Definition, Uses and Limitations of Optimization method.The Linear Programming Problem: Introduction, Formulation Of LPP, Graphical Solution And Some Exceptional Cases, Canonical And Standard Form Of LPP.

The Simplex Method: Solution of LPP By Simplex Method, Exceptional Cases, Artificial Variable Techniques (Big M), Two Phase Of Simplex Method, Problem of Degeneracy.

UNIT-II

The Dual Simplex Method: Dual And Primal Problem, Duality And Simplex Method, dual simplex method,Revised Simplex Method, Solution Of LPP Using Revised Simplex Method.

Networking Scheduling By PERT/CPM: Introduction, Basic Concepts, Constraints In Network, Construction Of The Network, Time Calculation In Networks, Critical Path Method (CPM), PERT, PERT Calculation, Advantage Of Network (PERT/CPM).

UNIT-III

The Transportation Problem: Introduction, Basic Feasibility Solution, Standard Transportation Problem, Balanced Transportation Problem, Multicommodity Transportation Problem, Row Minimum, Column Minimum, Matrix Minimum Method, Vogel Approximation Method (VAM), Optimality In Transportation Problem, (stepping stone and modified distribution methods) Degeneracy In Transportation Problem, Assignment And Routing Problem.

UNIT-IV

Game theory: Significance, essential features and limitations; Maximax and minimax principle, Game with pure & mined strategies, sul-game method (case of 2xn or mx2 methods), Probability method, graphic method, algebraic method.

Inventory Control: Introduction, Inventory Control, Selective Control Techniques, ABC Analysis Procedure, Economics Lot Size Problems, Problem of EQQ With shortage, Inventory Control Techniques Uncertain Demand, Stochastic Problems.

Text Book:

Kanti Swarup, P.K. Gupta and Manmohan, “Operations Research”, Sultan Chand & Sons. New Delhi.

Reference Books:

1. H. A. Taha, “Operation Research - An Introduction”, Macmillan Publications.
2. S. D. Sharma, “Operation Research”, Kedar Nath Ram Nath & Company, Merut.
3. K. K. Chawla, Vijay Gupta, Bhushan K Sharma, “Operations Research: Quatization Analysis for Management”, Kalyani Publishers, Kolkata.
4. V .K. Kapoor, “Operation Research”, Sultan Chand & sons, New Delhi.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand Fundamentals of Optimization Techniques.
CO2: Imply different Scheduling Methods.
CO3: Cast engineering minima/maxima problems into optimization framework.
CO4: Learn efficient computational procedures to solve optimization problems.
CO5: Use Matlab to implement important optimization methods.
CO6: Analyze Transportation Problems and its Concepts.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-14: Data Structure & Algorithm Analysis L:4 T:0 P:0

Course Objectives: This course provides a comprehensive understanding of algorithms and data structures essential for efficient problem-solving in computer science. It covers asymptotic analysis, algorithm design techniques, various data structures, sorting and searching algorithms, graph algorithms, and string matching algorithms. Students will learn to analyze algorithm performance, design efficient algorithms, and implement data structures to solve real-world problems.

UNIT-I

Analysis of algorithms: asymptotic notation, best case, worst case and average case analysis of algorithm.

Algorithm Design Techniques: Divide-and-conquer, Dynamic Programming, Greedy Method, Backtracking, Branch-and-bound.

Data Structures: Importance and need of good data structures, Arrays, memory representation and various operations on arrays, Multidimensional arrays, Sequential allocation, Address calculation, Sparse arrays. List: Simple Array Implementation of Lists, Linked Lists, Doubly Linked Lists, Circularly Linked list. Stack: Implementation and Applications of Stacks, Strategies for choosing the appropriate data structures.

UNIT-II

Queue, Applications of Queues, Array & Linked list implementation of Queues, Circular Queues, Deque.

Trees: Implementation of Trees, Tree Traversal, Binary Trees, Types of Binary Tree, Expression trees, Binary Search Tree (BST) & its implementation, Operations on BST, AVL Trees & operations on AVL trees, Single & Double Rotation in AVL trees, B-trees, Red-Black trees, Splay trees.

UNIT-III

Priority Queues and its Implementation, Binary Heap & its implementation, Heap Order Property, Basic Heap Operation, Application of Priority Queues- The Selection Problem, Event Simulation.

Sorting & Searching: Insertion Sort, Shell sort, Heap sort, Merge sort, Bubble sort, Quick sort, Bucket Sort, Radix sort, Linear search, Binary search, Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

UNIT-IV

Graphs & Algorithms: Representation, Type of Graphs, Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism, Graph Coloring, Covering and Partitioning, Depth- and breadth-first traversals, Minimum Spanning Tree: Prim’s & Kruskal’s algorithms, Shortest-path Algorithms: Dijkstra’s and Floyd’s algorithm, Topological sort, Max flow: Ford-Fulkerson algorithm, max flow – min cut.

String Matching Algorithms: Suffix arrays, Suffix trees, Rabin-Karp, Knuth-Morris-Pratt, Boyer-Moore algorithm Huffman Codes, Closets-Points Problem, The Selection Problem, Ordering Matrix Multiplications, Optimal Binary Search Tree.

Text Books:

1. Narasimha Karumanchi: “Data Structures and Algorithms Made Easy”, CarrerMonk 2016.
2. R. Panneerselvam, “Design and Analysis of Algorithms”, PHI 2nd edition 2016.
3. Thomas Coremen, “Introduction to Algorithms”, 3rd edition, Prentice Hall of India 2009.

Reference Books:

1. Seymour Lipschutz: "Data Structure With C", McGraw Hill, 1 Jan 2017.
2. Alfred, Hopcroft, Ullman (AHO): "The Design and Analysis of Algorithms", Pearson 2009.
3. Langsam / Augenstein / Tenenbaum: "Data Structures Using C and C++", 2nd edition, Pearson 2015.
4. Yashvant Kanetkar: "Data Structures Through C", 4th edition, bpb publisher 2022.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand the basics of algorithm design techniques and develop algorithms for Linked List, Stack data structures.
CO2: Develop algorithms for Queue, Binary Tree, Binary Search Tree, AVL tree, Red-black tree.
CO3: Develop algorithms for Priority Queues, Heaps, Hashing and sorting.
CO4: Acquire the knowledge of graphs, minimum spanning tree and string-matching algorithms.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-15(i): Operating system and Case Study **L:4 T:0 P:0**

Course Objectives: This course provides a thorough exploration of operating systems, encompassing their functions, various types, as well as key components such as process management, memory management, file systems, protection mechanisms, secondary storage management, security, distributed systems, and case studies with an emphasis on Linux. Students will delve into the foundational principles, algorithms, and methodologies essential for designing and building contemporary operating systems. Additionally, they will gain insight into pivotal security principles and approaches.

UNIT-I

Introduction: Introduction, Operating System Functions (Process Management, Memory Management, Storage Management, Protection and Security), Different types of Operating systems (Batch, Multiprogrammed, Time sharing, Real time, Distributed, Parallel), System calls and its types, System program.

Process Management: Introduction to Processes (Concept of processes, Process scheduling, Operations on processes), Interprocess Communication (IPC) (Shared Memory systems, Message passing systems), Critical Sections, Mutual Exclusion with Busy Waiting, Sleep and Wakeup, Semaphores (Binary Semaphore, Counting Semaphore), CPU Scheduling (Scheduling criteria, pre- emptive & non-pre-emptive scheduling), Scheduling Algorithms (FCFS, SJF, RR, Priority Scheduling, Multilevel queue scheduling, Multilevel feedback queue scheduling, Multiple processor scheduling, Real time scheduling).
Threads: Overview of threads, Multicore Programming, Multithreading Models, Thread libraries, Implicit Threading, Threading Issues.

UNIT-II

Deadlocks: Introduction to deadlocks, Conditions for deadlock (Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait), Resource allocation graphs, Deadlock Detection and Recovery, Deadlock Prevention, Deadlock Avoidance (Banker's Algorithm).

Memory Management: Background, Logical vs. Physical address space, Memory management without swapping, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand paging, Performance, Page replacement, Page replacement algorithms (FIFO, Optimal, LRU, LRU-approximation, Counting based Page replacement), Thrashing (Causes, Working-set model, Page-fault frequency).

File Systems: Files (Concept, File structure, File types, Access methods, File attributes, File operations), Directory structure, Allocation methods (contiguous, linked, indexed), Free- space management (bit vector, linked list, grouping), File Protection mechanisms.

UNIT-III

Protection: Goals, Principles, Domain Protection, Access Matrix and its Implementation, Access Control, Revocation of access rights, Language Based Protection (Compiler-based enforcement, Protection in Java).

Secondary Storage: Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK), Disk Management (Disk Formatting, Boot Blocks, Bad Blocks), Swap Space Management (Swap Space use, Swap Space Location, Swap Space Management).

Security: Security Problems, Program Threats (Trojan horse, Trap Door, Logic Bomb, Stack and buffer overflow, Viruses), System and Network threats (Worms, Port Scanning, Denial of Service), Cryptography (Encryption: Symmetric, Asymmetric, Authentication, Key distribution), Implementation, User authentication (Passwords, Vulnerabilities, Securing passwords, One-time Passwords, Biometrics).

UNIT-IV

Distributed System: Types of Network-based OS, Network structure and topologies, Communication structure & Protocol, Design issues, Distributed File System (Remote File Access, File replication), Distributed synchronization (Mutual Exclusion, Concurrency control, Deadlock handling).

Case Studies: Brief introduction and History of LINUX, Design Principles and Components, Kernel Modules (Module Management, Driver Registration, Conflict Resolution), Process Management (fork() and exec() Process Model, Process and Threads), Scheduling (Process scheduling, Real-Time Scheduling, Kernel Synchronization, Symmetric Multiprocessing), Memory Management (Physical Memory, Virtual Memory, Execution and loading of user Programs), File Systems (Virtual File system, LINUX ext3 file system, Journaling, LINUX Process File system), Input and Output (Block devices, Character Devices), Interprocess Communication (Synchronization and signals, Passing of Data among Processes) Network Structure.

Text Books:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. "Operating System Concepts", Wiley Publishing.

Reference Book:

1. William Stallings, "Operating Systems", Macmillan Publishing Company.
2. Deitel H.M., "An Introduction to Operating System", Addison Wesley Publishing Company, 2984.
3. Tanenbaum, A.S., "Modern Operating System", Prentice Hall of India Pvt. Ltd.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand fundamental concepts related to operating system, processes, CPU scheduling algorithms, Inter Process Communication, Process synchronization, Critical Sections, Semaphores and threads.
CO2: Identify conditions leading to deadlocks and apply deadlock prevention and avoidance techniques.
CO3: Analyses different memory management techniques as well as file system structures and its allocation methods.
CO4: Evaluate page replacement algorithms and techniques to handle thrashing in memory management.
CO5: Understand the goals and principles of protection mechanisms in operating systems.
CO6: comprehend secondary storage management including disk structure, disk scheduling, and swap space management.
CO7: Analyze security issues in operating systems, including program threats, system/network threats, and cryptography.
CO8: Understand the concept of distributed systems.
CO9: Analyze case studies of operating systems like LINUX, including kernel modules, process management, memory management, file systems and Inter-process communication in a distributed environment.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-15(ii): Advanced Software Engineering Concepts L:4 T:0 P:0

Course Objectives: This course serves as an introduction to the fundamental principles and methodologies of software engineering. It covers the software development process, various software life cycle models, standards for developing these models, object-oriented methodology, system architecture, UML modeling, system design, and testing of object-oriented systems. Students will gain a comprehensive understanding of the software engineering lifecycle, from requirements gathering to testing, while emphasizing the importance of adherence to established standards and methodologies.

UNIT-I

Introduction to Software Engineering: Software Engineering Development, Software Engineering Development, Software Life Cycle Models, Standards for developing life cycle models.

UNIT-II

Object Methodology & Requirement Elicitation: Introduction to Object Oriented Methodology, Overview of Requirements Elicitation, Requirements Model-Action & Use cases, Requirements Elicitation Activities, Managing Requirements Elicitation.

UNIT-III

Architecture: Model Architecture, Requirements Model, Analysis Model, Design Model, Implementation Model, Test Model.

Modeling with UML: Basic Building Blocks of UML, A Conceptual Model of UML, Basic Structural Modeling, UML Diagrams.

System Analysis: Analysis Model, Dynamic Modelling & Testing.

UNIT-IV

System Design: Design concepts & activities, Design models, Block design, Testing.

Testing Object Oriented Systems: Introduction, Testing Activities & Techniques, The Testing Process, Managing Testing Case Studies.

Text Books:

1. Stephen R. Scach, “Classical & Object Oriented Software Engineering with UML and Java”, McGraw Hill, 1999.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Identify and correct typical requirements quality issues.
CO2: Apply different testing, code review, code analysis, and code refactoring approaches.
CO3: Explain industrial state of the practice methods of advanced software engineering.
CO4: To apply the knowledge of the software engineering to actual software development problems including at the industries.
CO5: To analyze source code, identify code smell and improve existing software by apply the knowledge of software engineering to actual the software engineering problems.
CO6: To analyze software requirement for industrial applications.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-15(iii): Cyber Law **L:4 T:0 P:0**

Course Objectives: This course provides an in-depth exploration of cyber law, covering its definition, scope, socio-legal implications, and various aspects related to cyber crimes and regulations. Students will gain a comprehensive understanding of the legal framework surrounding computer science, including contractual aspects, liability issues, and the Information Technology Act, 2000, with a focus on digital signatures and electronic governance.

UNIT-I

Cyber Law: Introduction, Definition, nature & Scope of Cyber Laws. Sociolegal Implications of Computer Science, Cyber Laws. Cyber Crimes: Definition & Kinds of Cyber Crimes. International and Foreign Developments. Common Cyber Offences: Pheakering, Internet Frauds, Hackers, Stalking, E-Mail, Security Invasion, Money Laundering, Data Diddling, Theft of Information.

UNIT-II

Contractual Aspects: Hardware Contracts: User Requirement Specification, Negotiation, Sales & Leases, Delivery & Payment, Seller's Obligations, Buyer's Remedies.

Software Contract: Selecting Software, Types of Software, What is Software, Software License, Principal Commercial Terms, Warranties, Software Maintenance.

Liability: Contractual Liability, Strict Liability, Negligence, Criminal. Miscellaneous (Briefly); Copyright & Patent Protection, Evidence, Protecting Confidential Information.

UNIT-III

The Information Technology Act, 2000: Introduction: Definition, A Brief Summary of the Act. Digital Signature & Electronic Governance (Sections 3 to 10) Secure Electronic Records & Secure Digital Signatures (Sections 14 to 16).

UNIT-IV

Regulation of Certifying Authorities (Sections 17 to 34). Digital Signature Certificates (Sections 35 to 39). Duties of Subscribers (Sections 40 to 42). Penalties, Adjudication Offences (Sections 45 to 47 & Sections 65 to 78). Cyber Regulations Appellate Tribunal (Sections 48 to 64).

Text Books:

1. The Information Technology Act, 2000.

Reference Books:

1. Chris Reed (Ed.), Computer Law, 1996: Universal Law Publishing Co. Pvt. Ltd.
2. Mittal D.P., Law of Information Technology (2000): Taxmann's.

Course Outcomes:

By the end of this course, the student will be able to:
CO1: Understand Cyber Space, Cyber Crime, Information Technology, Internet & Services.
CO2: List and discuss various forms of Cyber Crimes.
CO3: Explain Computer and Cyber Crimes.
CO4: Understand Cyber Crime at Global and Indian Perspective.
CO5: Describe the ways of precaution and prevention of Cyber Crime as well as Human Rights.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-21: Research Methodology in Computer Science L:4 T:0 P:0

Course Objectives: This course is designed to equip students in computer science with the essential skills and knowledge needed to conduct rigorous and effective research. It covers various research methodologies, techniques, and ethical considerations relevant to the field.

UNIT-I

Introduction to Research Methodology: Overview of research in computer science, Types of research: Basic, Applied, and Experimental, Research paradigms: Positivism, Interpretivism, and Pragmatism, Formulation of research problems and objectives.
Literature Review: Importance of literature review in research, Conducting a systematic literature review, Critical analysis and synthesis of existing research, Identifying research gaps.

UNIT-II

Research Design and Planning: Experimental design, Survey design, Case study design, Mixed-methods research, Sample selection and size determination. Data Collection Methods: Quantitative data collection techniques (Surveys, experiments, observations), Qualitative data collection techniques (Interviews, focus groups, case studies), Ethical considerations in data collection.

UNIT-III

Data Analysis Techniques: Quantitative data analysis using statistical tools (Descriptive Statistics, Inferential Statistics, Data Visualization, Statistical Software), Qualitative data analysis techniques (Thematic analysis, content analysis, grounded theory), Computer-assisted data analysis tools (Programming Languages and Libraries, Database Management Systems, Statistical Software, Data Visualization Tools). Emerging Trends in Computer Science Research and Advanced Data Analysis Techniques: Exploration of current research trends and hot topics, Overview of advanced data analysis techniques in computer science research, Introduction to machine learning for data analysis, Big data analytics and tools in computer science research (Hadoop, Apache Spark, NoSQL Databases, Apache Flink, Tableau).

UNIT-IV

Significance of Report Writing: Different Steps in writing Report, Layout of the Research Report, Types of Reports, Mechanics of Writing a Research Report, Art of scientific writing- Steps to better writing, flow method, organization of material and style, Drawing figures, graphs, tables, footnotes, references etc. in a research paper. Use of Internet in Research Work: Use of internet networks in research activities.

References:

1. Creswell, J. W. (2014). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.
2. Wallwork, A (2016). English for Writing Research Papers.
3. Booth, W. C., Colomb, G. G., & Williams, J. M. (2008). The Craft of Research.
4. Silverman, D. (2016). Qualitative Research.

Course Outcomes:
By the end of this course, the student will be able to:
Co1: demonstrate proficiency in conducting a systematic literature review.
CO2: understand and apply various research paradigms and methodologies relevant to computer science.
CO3: formulate research problems and objectives based on a solid understanding of research design principles.
CO4: apply quantitative and qualitative data analysis techniques.
CO5: develop effective communication skills for presenting research findings.
Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-22: Research Ethics **L:4 T:0 P:0**

Course Objectives: This course aims to familiarize students in computer science with ethical principles and guidelines governing research conduct. It explores the unique ethical challenges within the field and equips students with the knowledge and skills to conduct research responsibly and with integrity.

UNIT-I

Introduction to Research Ethics: Overview of research ethics, Historical perspectives and landmark ethical cases, The role of ethics in computer science research; Ethical Principles in Research: Fundamental ethical principles (e.g. autonomy, beneficence, justice), Application of ethical principles in computer science research, Case studies illustrating ethical dilemmas in research.

UNIT-II

Informed Consent and Human Subjects Research: Importance of informed consent, Guidelines for obtaining informed consent in human subjects research, Ethical considerations in human-computer interaction studies; Privacy and Data Protection: Privacy concerns in computer science research, Responsible handling of sensitive data, Legal and ethical aspects of data protection.

UNIT-III

Intellectual Property and Plagiarism: Understanding intellectual property rights, Ethical use of software, algorithms, and data-sets, Strategies to avoid plagiarism and maintain academic integrity; Emerging Ethical Issues in Computer Science: Ethical considerations in artificial intelligence and machine learning, Autonomous systems and ethical decision-making, Ethical implications of Cyber security research.

UNIT-IV

Research Collaboration and Authorship: Ethical collaboration in interdisciplinary research, Guidelines for authorship and contributor-ship, Addressing conflicts of interest in collaborative projects; Responsible Research Communication: Ethical publication practices, Peer review and ethical responsibilities of reviewers, Ethical use of social media and online platforms for research communication.

References:

1. Shamoo, A. E., & Resnik, D. B. (2015). Responsible Conduct of Research.
2. Steneck, N. H. (2007). Introduction to the Responsible Conduct of Research.
3. Macrina, F. L. (2014). Scientific Integrity: Text and Cases in Responsible Conduct of Research.

Course Outcomes:
By the end of this course, the student will be able to:
CO1: understand the Fundamental Ethical Principles such as autonomy, beneficence, and justice, and their application in the context of computer science research.
CO2: apply ethical principles to address specific challenges and dilemmas that arise in computer science research, including issues related to privacy, data protection, intellectual property, and emerging technologies.
CO3: gain competence in obtaining informed consent, addressing privacy concerns, and ensuring ethical conduct in research involving human subjects, with a particular emphasis on the unique considerations in human-computer interaction studies.
CO4: develop an awareness of and sensitivity to emerging ethical issues in computer science, including those related to artificial intelligence, machine learning, autonomous systems, and cyber security.
CO5: understand the importance of responsible research conduct, including issues related to authorship, collaboration, and responsible communication of research findings.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-23: Distributed Data Base Management System L:4 T:0 P:0

Course Objectives: This course offers a thorough exploration of distributed data processing and distributed database management systems, providing students with a comprehensive understanding of their core principles and functionalities. It encompasses the fundamental aspects of DDBMS, Additionally, the curriculum covers various facets of relational database management systems, including data modeling, normalization techniques, distributed database architecture, design methodologies, query optimization, distributed concurrency control mechanisms, and ensuring reliability within distributed DBMS environments.

UNIT-I

Distributed Data Processing: Introduction, Fundamentals of Distributed Data Base Management System (Transparent management of distributed & replicated data, Reliability, Improved performance, System expansion), Disadvantages of Distributed Data Base Management System (Complexity, Cost, Distribution of control, Security, Distributed database design, Query processing, Directory Mgmt, concurrency control, Deadlock Mgmt, Reliability, OS support, Heterogeneous databases, Relationship).

Relational Data Base Management System: Basic Concepts, Data Modeling for a Database, Records and Files, Abstraction and Data Integration, The Three- Level Architecture Proposal for DBMS, Components of a DBMS, Advantages and Disadvantages of a DBMS. Data Models, Data Associations, Data Models Classification, Entity Relationship Model, Relational Data Model.

Normalization: Dependency structures, Normal forms.

UNIT-II

Distributed Data Base Management System Architecture: Architectural models for distributed DBMS (Autonomy, Distribution, Heterogeneity, Architectural alternatives), Client/server systems, Peer-to-peer Distributed Systems.

Distributed Database Design: Design Strategies (Top-Down Design & Bottom- Up design process), Design issues (reasons for fragmentation, alternatives, Degree & Correctness rules of fragmentation, Allocation alternatives, Information requirement.

Fragmentation: Horizontal, Vertical, Hybrid Fragmentation. Allocation: Problem, Information requirement, Allocation model, Solution methods.

UNIT-III

Query Processing: Problem, objectives, Complexity of Relational Algebra operations, Characterization of query processing (Language, Types of Optimization, Optimization timing, Statistics, Decision sites, Exploitation of network topology & Replicated fragments, Use of semi joins), Layers of Query processing (Query decomposition, Data localization, Global & Local query optimizations).

Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanism, Locking based concurrency control algorithm (centralized 2pl, primary copy 2pl, distributed 2pl), Timestamp based concurrency control algorithm (conservative & multiversion to algorithm), Optimistic concurrency control algorithm, Deadlock management, prevention, avoidance, detection & resolution.

UNIT-IV

Distributed DBMS Reliability: Reliability concepts & measures (system, state & failures, reliability & availability, mean time between failures/repair), Failures & fault tolerance in distributed system (reason for failures, fault tolerance approaches & techniques), Failures in Distributed DBMS (transaction, system, media & communication failure), Local reliability protocols (architectural considerations, recovery, information execution of LRM commands, checkpointing, handling media failure), Distributed Reliability Protocols (Components, Two-Phase commit protocol, Variation of 2PC).

Text Books:

1. M. Tamer Ozsu & Patrick Valduriez, “Principles of Distributed Database Systems”, Pearson Education Asia.
2. Desai, B., “An Introduction to Database Concepts.” Galgotia Publications, New Delhi.

Reference Book:

1. Date C.J., “An Introduction to Database Systems”, Narosa Publishing House, New Delhi.
2. Elimsari and Navathe, “Fundamentals of Database Systems”, Addison Wesley, New York.
3. Ullman, J.D, “Principals of Database Systems”, Galgotia Publications, New Delhi.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand concepts of Distributed Data Processing.
CO2: Gain knowledge about relational database management system.
CO3: Gain insight into the architectural models for distributed DBMS.
CO4: Understand query processing objectives and explore distributed concurrency control mechanisms.
CO5: Understand reliability concepts and analyze failures and fault tolerance approaches and techniques in distributed systems.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-24: Machine Learning using Python **L:4 T:0 P:0**

Course Objectives: This course provides a comprehensive introduction to machine learning (ML) using Python, covering essential concepts, techniques, and libraries. Students will gain an understanding of the motivations behind machine learning, the types of problems it can solve, and the importance of understanding both the task at hand and the underlying data. Through practical examples, students will become proficient in using key libraries such as NumPy, pandas, matplotlib, SciPy, and scikit-learn.

UNIT-I

Introduction to Machine Learning with Python: Introduction to Machine Learning, Why Machine Learning? Problems Machine Learning Can Solve, Knowing Your Task and Knowing Your Data, Essential Libraries and Tools: NumPy, pandas, matplotlib, SciPy scikit-learn.

UNIT-II

Supervised Learning with Python: Classification and Regression , Generalization, Overfitting, and Underfitting, Relation of Model Complexity to Dataset Size ,Supervised Machine Learning Algorithms: (k-Nearest Neighbors, Linear Models, Naive Bayes Classifiers, Decision Trees, Ensembles of Decision Trees, Support Vector Machines, Uncertainty Estimates from Classifiers, The Decision Function, Predicting Probabilities, Uncertainty in Multiclass Classification.

UNIT-III

Unsupervised Learning and Preprocessing with Python: Types of Unsupervised Learning, Challenges in Unsupervised Learning, Preprocessing and Scaling, Different Kinds of Preprocessing, Applying Data Transformations, Scaling Training and Test Data the Same Way, Dimensionality Reduction, Feature Extraction, and Manifold learning: Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF), Manifold Learning with t-SNE Clustering, k-Means Clustering, Agglomerative Clustering, DBSCAN, Comparing and Evaluating Clustering Algorithms.

UNIT-IV

Representing Data, Engineering Features and Model Evaluation: Categorical Variables, One-Hot-Encoding (Dummy Variables), Binning, Discretization, Linear Models, and Trees, Interactions and Polynomials, Univariate Nonlinear Transformations, Automatic Feature Selection, Utilizing Expert Knowledge, Model Evaluation and Improvement, Cross-Validation, Grid Search, Evaluation Metrics and Scoring, Algorithm Chains and Pipelines, Parameter Selection with Preprocessing, Building Pipelines Working with Text Data, Types of Data Represented as Strings, Example Application: Sentiment Analysis of Movie Reviews, Representing Text Data as a Bag of Words.

Text Books:
1. Introduction to Machine Learning with Python: A Guide for Data Scientists (Greyscale Indian Edition) by Andreas Muller & Sarah Guido, O'Reilly Media, 2016.
Reference Book:
1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, by Aurélien Géron, O'Reilly Media, Second Edition, 2019. 2. Applied Machine Learning: 2nd Edition, by M. Gopal McGraw Hill Education, 2018.

Course Outcomes:
By the end of this course, the student will be able to:
CO1: Understand fundamental concepts and applications of machine learning.
CO2: Analyze various supervised learning techniques and implement machine learning system design.
CO3: Identify different data analysis techniques like frequent pattern analysis, classification, and clustering.
CO4: Demonstrate the use of various machine learning techniques on different application data sets.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-25(i): Artificial Intelligence & Expert System **L:4 T:0 P:0**

Course Objectives: This course provides a comprehensive introduction to the field of Artificial Intelligence (AI), covering its history, importance, and various fundamental concepts and techniques. Students will explore intelligent agents, problem-solving methods, knowledge representation and manipulation, formalized symbolic logics, expert systems, and applications of AI in different domains. Additionally, the course will address ethical considerations and future trends in AI.

UNIT-I

Introduction of AI, History and Importance of AI, Intelligent agents: Types, architectures, and environments, Problem solving: Techniques, search algorithms (uninformed and informed), game playing. Knowledge, Knowledge-Based Systems, Representation of Knowledge, Knowledge Organization, Knowledge Manipulation, Acquisition of Knowledge.

UNIT-II

Formalized Symbolic Logics Syntax and Semantics For Propositional Logic, Properties of Wffs, Conversion To Clausal Form, Inference Rules, Resolution. Dealing With Inconsistencies - Truth Maintenance Systems, Symbolic Reasoning under Uncertainty, Statistical Reasoning. Structural Knowledge Graph, Frames and Related Structures.

UNIT-III

Introduction to expert systems: Architecture, components, knowledge acquisition. Knowledge engineering: Techniques for knowledge acquisition, knowledge representation for expert systems. Explanation facilities: Justification, reasoning behind conclusions. Development tools: Expert system shells, case-based reasoning.

UNIT-IV

Applications of AI: Natural language processing, computer vision, robotics. Applications of expert systems: Medical diagnosis, financial decision making, engineering design. Ethical considerations of AI: Bias, transparency, safety. Future trends in AI: Deep learning, artificial general intelligence.

Text Books:

1. Introduction to Artificial Intelligence and Expert Systems by Dan W. Patterson (Pearson Education).
2. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig (Pearson Education).

Reference Book:

1. Artificial Intelligence by Elaine Rich, Kevin Knight, and Shivashankar B Nair (Tata McGraw Hill).
2. Expert Systems: Principles and Programming by Giarrantana/Riley (Thomson).

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Define and explain key concepts in Artificial Intelligence.
CO2: Apply problem-solving techniques like search algorithms and game playing.
CO3: Represent knowledge using various formalism like logic and semantic networks.
CO4: Employ reasoning techniques like forward and backward chaining.
CO5: Analyze the strengths and weaknesses of different learning algorithms.
CO6: Design and develop basic expert systems for specific domains.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-25(ii): Big Data Analytics **L:4 T:0 P:0**

Course Objectives: This course offers a thorough introduction to big data analytics, encompassing crucial principles, technologies, and tools needed for the analysis of vast datasets. Students will explore parallel processing methods necessary for managing large-scale data, along with designing data architecture, storage, and analysis techniques. Furthermore, the course will examine diverse applications and real-world case studies of big data analytics.

UNIT-I

Introduction: Introduction to Big Data Analytic, Big Data, Scalability and Parallel Processing, Designing Data Architecture, Data Sources, Quality, Pre-Processing and Storing, Data storage and Analysis, Big Data Analytics Applications and Case Studies.

Introduction to Hadoop: Hadoop and its Ecosystem, Hadoop Distributed File System, Mapreduce Framework and Programming Model, Haddop yarn, Hadoop Ecosystem Tools.

UNIT-II

NoSQL Big Data Management: Introduction to NoSQL, NoSQL Data Store, NoSQL Data Architecture Patterns, NoSQL to Manage Big Data, Shared-Nothing Architecture for Big Data Tasks; MongoDB Database.

UNIT-III

MapReduce, Hive and Pig: Introduction, MapReduce Map Tasks, Reduce Tasks and MapReduce Execution, Composing MapReduce for Calculations and Algorithms, Hive, HiveQL, Pig.

UNIT-IV

Spark and Big Data Analytics: Introduction, Spark, Introduction to Data Analysis with Spark, Downloading Spark, and Programming using RDDs and MLIB, Data ETL (Extract, Transform and Load) Process, Introduction to Analytics, Reporting and Visualizing.

Text Books:

Raj Kamal and Preeti Saxena, BIG DATA ANALYTICS: Introduction to Hadoop, Spark, and Machine-Learning. N.p., McGraw-Hill Education, 2019.

Reference Books:

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
2. P. J. Sadalage, M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2012.
3. Tom White, Hadoop: The Definitive Guide, 3/e, O'Reilly, 2012.

Course Outcomes:
By the end of this course, the student will be able to:

CO1: Understand Spark, Hive, Pig, NoSQL.
CO2: Apply big data management techniques to generate different patterns.
CO3: Analyze Big Data and its business implications.
CO4: Evaluate data using different tools. CO5: Create mechanisms for reporting and visualizing the big data.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-25(iii): Cloud Computing **L:4 T:0 P:0**

Course Objectives: This course offers a comprehensive exploration of cloud computing, covering its definition, characteristics, components, and various cloud service models such as Software as a Service (SAAS), Platform as a Service (PAAS), and Infrastructure as a Service (IAAS). Students will examine organizational scenarios of cloud usage, administration, monitoring of cloud services, as well as the benefits and limitations of cloud computing.

UNIT-I

Introduction to Cloud Computing, Definition, Characteristics, Components, Cloud provider, SAAS, PAAS, IAAS and Others, Organizational scenarios of clouds, Administering & Monitoring cloud services, benefits and limitations, Deploy application over cloud, Comparison among SAAS, PAAS, IAAS Cloud computing platforms: Infrastructure as service: Amazon EC2, Platform as Service: Google App Engine, Microsoft Azure, Utility Computing, Elastic Computing.

UNIT-II

Roots of SOA : Characteristics of SOA - Comparing SOA to client-server and distributed internet architectures - Anatomy of SOA- How components in an SOA interrelate - Principles of service orientation.

Web Services: Service descriptions- Messaging with SOAP- Message exchange; Patterns- Coordination- Atomic Transactions- Business activities- Orchestration- Choreography- Service layer abstraction- Application Service Layer- Business; Service Layer- Orchestration Service Layer.

Service Oriented Analysis: Business-centric SOA- Deriving business services- service modeling- Service Oriented Design- WSDL basics- SOAP basics- SOA composition guidelines- Entity-centric business service design- Application service design- Task centric business service design.

UNIT-III

Cloud Technology: Introduction to Cloud Technologies, Study of Hypervisors Compare SOAP and REST Webservices, AJAX and mashups-Web services: SOAP and REST, SOAP versus REST, AJAX: asynchronous 'rich' interfaces, Mashups: user interface services.

Virtualization Technology: Virtual machine technology, virtualization applications in enterprises, Pitfalls of virtualization.

Multitenant software: Multi-entity support, Multi-schema approach, Multi- tenance using cloud data stores, Data access control for enterprise applications, Data in the cloud: Relational databases, Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo.

UNIT-IV

Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud Cloud computing security architecture: Architectural Considerations- General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro-architectures; Identity Management and Access control-Identity management, Access control, Autonomic Security.

Cloud Computing Security Challenges: Virtualization security management- virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud.

Text Books:

1. Cloud Computing for Dummies by Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper (Wiley India Edition)
2. Enterprise Cloud Computing by Gautam Shroff, Cambridge.

3. Thomas Erl, "Service Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.
4. Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley-India.

Reference Books:

1. Google Apps by Scott Granneman, Pearson.
2. Cloud Security & Privacy by Tim Malhar, S.Kumaraswammy, S. Latif (SPD,O'REILLY).
3. Cloud Computing: A Practical Approach, Antohy T Velte, et.al McGraw Hill.
4. Cloud Computing Bible by Barrie Sosinsky, Wiley India.
5. Stefano Ferretti et.al., QoS–aware Clouds", 2010 IEEE 3rd International Conference on Cloud Computing

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
CO2: Apply the fundamental concepts in data-centers to understand the trade-offs in power, efficiency and cost.
CO3: Identify resource management fundamentals, i.e. resource abstraction, sharing and sand-boxing and outline their role in managing infrastructure in cloud computing.
CO4: Analyze various cloud programming models and apply them to solve problems on the cloud.
CO5: Learn the Concept of Cloud Infrastructure Model.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-31: Computer Networks **L:4 T:0 P:0**

Course Objectives: This course provides a comprehensive understanding of data communication and networking, covering essential concepts, protocols, and standards. Students will learn about network components, topologies, transmission modes, and categories of networks, along with the OSI Reference Model and TCP/IP reference model. The course explores the physical layer, data link layer, network layer, transport layer, and application layer, addressing various design issues, protocols, and algorithms.

UNIT-I

Data Communication, Network Components, Protocol & Standards, Standard Organization, Topologies, Transmission modes, Categories of Networks, Uses, Applications.
The OSI Reference Model: Layered architecture, Functions of layers, TCP/IP reference model, Comparison of OSI & TCP/IP models, addressing schemes.
Physical layer: analog and digital signals, causes of transmission impairment, bandwidth, throughput and latency, Multiplexing, Guided and wireless transmission media, switching techniques, Communication satellites, Modems, Digital Subscriber Line.

UNIT-II

Data Link and Mac Layer: Design issues, Framing techniques, Flow control, Error Control, Error Detecting code and Error Correcting codes, Data link Control: Protocols for noiseless channel-- Simplest Protocol, Stop-and-Wait Protocol, For Noisy Channel-- Stop-and-Wait ARQ, Go-Back-N ARQ, and Selective-Repeat ARQ Protocol, HDLC Protocol, and PPP Protocol, Multiple Access (Random & Controlled)-- ALOHA, CSMA, CSMA/CD, CSMA/CA, Reservation, Polling, Token passing, Channelization--FDMA, TDMA, CDMA, and IEEE standards-- 802.3 (Ethernet), 802.4 (Token Bus), 802.5 (Token Ring), 802.11(Wireless LAN), 802.15 (Bluetooth).

UNIT-III

Network and Transport Layer: Network layer design issues, IPv4 & IPv6 Addressing, Routing algorithms-shortest path routing, flooding, distance vector routing, link state routing, hierarchical routing, broadcast routing, multicast routing, routing for mobile hosts, Congestion Control algorithms, congestion prevention policies, congestion control in virtual circuit & datagram subnetworks, Quality of Service (QoS), Internetworking--Tunneling, internetwork routing, fragmentation, Network layer in Internet IP protocol, IP Address, OSPF, BGP, Internet multicasting, Mobile IP, Transport Layer: Concept of transport service, elements of transport protocols, transport protocol—TCP & UDP, Remote procedure call, Performance issues in computer networks.

UNIT-IV

Application Layer: services, protocols-- DNS, SMTP, FTP, TELNET, HTTP, WWW, Network Security: Cryptography concept and techniques, Symmetric Key algorithms—AES, DES, Asymmetric key algorithms-- RSA, Diffie Hellman, Message Digest and Hash algorithms, Digital Signature, IP security (IPSec) Firewalls. Internet radio, VoIP, PGP, Web security, social issues in network security.

Text Books:

1. B.A. Forouzan, “Data Communications and Networking”, 4th edition, Tata McGraw Hill 2017.
2. A.S. Tanenbaum, “Computer Networks”, 6th edition, Pearson 2022.

Reference Book:

1. J.F. Kurose & K.W. Ross, “Computer Networking”, 8th edition, Pearson 2022.
2. William Stallings, “Computer Networking with Internet Protocols and Technology”, 1st edition, Pearson 2023.
3. William Stallings, “Cryptography and Network Security - Principles and Practice”, 7st edition, Pearson 2017.
4. Atul Kahate, “Cryptography and Network Security”, 4th edition, Tata McGraw Hill 2019.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand the various transmission media and underlying architecture of Computer Network technologies.
CO2: Understand various issues during data transmission in wired/wireless technology and analyze different protocols to handle these issues.
CO3: Gain intuitive knowledge of different Routing algorithms and Transport layer protocols.
CO4: Learn different Application Layer protocols, get an insight into various network security issues and understand the cryptographic protocols, Digital Signature etc. to tackle these issues.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-32: Data Warehousing and Data Mining L:4 T:0 P:0

Course Objectives: This course offers a comprehensive understanding of data warehousing and decision support systems (DSS). It covers topics such as data warehouse architecture, system lifecycle, data sources, user requirements, conceptual modeling, logical modeling, data warehouse components, mapping data warehouse to a multiprocessor architecture, data tools, metadata, OLAP, and data mining techniques and algorithms.

UNIT-I

Introduction: DSS, Data warehouse Architecture, Data Staging & ETL, Multidimensional Model, Meta data, Accessing data warehouse, ROLAP, MOLAP, HOLAP.

System Lifecycle: Risk factors, Top-down, Bottom-up, Data mart design phases, Methodological framework, Testing data marts.

Data Sources: Inspecting and normalizing schemata, Integration problems, Integration phases, Mapping.

User Requirements & Conceptual Modeling: Glossary based requirements analysis, Goal-oriented requirements analysis, Dimensional Fact Model, Advanced modeling, Events and Aggregation, Time, Formalizing the dimensional fact model.

Conceptual Design: ER schema based design, Relational schema based design, XML schema based design, Mixed approach design.

UNIT-II

Logical Modeling & Design: MOLAP, HOLAP & ROLAP systems, Views, Temporal scenarios, Fact schemata to star schemata, View materialization, View Fragmentation, Populating - reconciled databases, dimension tables, fact tables & materialized views, Cleansing data.

Data Warehouse Components: Overall architecture, database, Sourcing, acquisition, cleanup and transformation tools, Metadata, Access tools, Administration and management, Info delivery System. Building a Data Warehouse: Considerations- business, design, technical & implementation, Integrated solutions, Benefits.

UNIT-III

Mapping Data Warehouse to a Multiprocessor Architecture: Relational database technology, Database architectures for parallel processing, Parallel RDBMS features and vendors.

DBMS Schemas & Decision Support: Data layout for best access, Multidimensional data models, Star schema.

Data Tools and Metadata: Tool requirements, Vendor approaches, Access to legacy data, Transformation engines, Metadata - definition, interchange initiative, repository, trends, Reporting & Query Tools – categories.

OLAP: Need, Multidimensional data model, guidelines, Multidimensional Vs multirelational OLAP, Categorization of OLAP tools.

UNIT-IV

Introduction: Data mining, Measuring effectiveness, Discovery Vs prediction, Overfitting, Comparing the technologies, Decision trees, Where to use them, General idea, How do they work, Strengths and weaknesses.

Techniques and Algorithms: Neural networks - uses, making predictions, different kinds, Kohonen feature map, their working, Nearest Neighbour & Clustering– uses, predictions and differences, their working, Genetic Algorithms– uses, cost minimization, cooperative strategies, their working, Rule

Induction– uses, evaluation of rules, rules Vs decision trees, their working, Using the right technique, Data mining & business process.

Text Books:

1. Data Warehousing, Data Mining & OLAP, Alex Berson & Stephen J. Smith, Tata McGraw-Hill, 2009.
2. Data Warehouse Design: Modern Principles and Methodologies, Matteo Golfarelli, Stefand Rizzi, Tata McGraw-Hill, 2009.

Reference Book:

1. Decision support and data warehouse systems, Efrem Mallach, Tata McGraw Hill, 2009.
2. The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems, John Wiley & Sons, 2008.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand the functionality of the various data mining and data warehousing component.
CO2: Appreciate the strengths and limitations of various data mining and data warehousing models.
CO3: Explain the analyzing techniques of various data.
CO4: Describe different methodologies used in data mining and data ware housing.
CO5: Compare different approaches of data warehousing and data mining with various technologies.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-33(i): Software Quality and Testing **L:4 T:0 P:0**

Course Objectives: This course provides a comprehensive understanding of software quality assurance and testing concepts, techniques, and methodologies. It covers topics such as software quality control, software quality assurance, software quality models, software quality metrics, software testing fundamentals, functional and non-functional testing, structural testing (white box testing), and levels of testing.

UNIT-I

Software and Quality Concept: Objectives, overview, Software perspective, Software Quality Control, Software Quality Assurance, Software Quality models, Software Quality measurement and metrics. Assuring Software Quality Assurance (SQA): Objectives, goals, responsibilities, life cycle, SQA planning, SQA monitoring and controlling, testing, setting standards and procedures, SQA techniques: Management Review Process, Walkthrough, Software inspection process, configuration audits & document verification, Developing and controlling relevant metrics, SQA activities- revision, process evaluation, software standards.

UNIT-II

Software Quality Metrics: Objectives, Software metrics, Software Quality metrics framework, Software Quality metrics features, Development of software quality metrics- SATC's approach, Kitchenham's approach, Abreu's approach, Victor's approach, Selection of software quality metrics- Size related metrics, complexity metrics, Halstead metrics, quality metrics. Software Quality Models: Objectives, Hierarchical model- factor- criteria metrics model, McCall's model, Boehm model, ISO 9126 model, Dromey's Quality model, Non-hierarchical model-Bayesian belief networks, star model, capability maturity models.

UNIT-III

Software Testing: Introduction, Definition (testing, fault, error, failure, bug, mistake), test oracle, test case, Process, Limitations of Testing, benefits & goals of testing. Functional Testing: Boundary Value Analysis- Introduction & Definition, Generalizing, limitations, Robustness testing, Worst case testing, Test cases. Equivalence Class Testing- Introduction & Definition, Weak normal, strong normal, Weak robust, Strong robust, Test cases. Decision Table Based Testing- Introduction & Definition, technique, test cases. Non-functional testing, Acceptance Testing: types, criteria, process.

UNIT-IV

Structural Testing (White Box Testing): Path testing - Introduction & definition, pitfalls & tools for structural testing. DD-path, Test coverage metrics, statement coverage, decision coverage, condition coverage, path coverage McCabe's basis path method, its observations and complexity. Control flow testing, Data Flow Testing: Definition, data flow graphs, data flow model, Data flow testing strategies. Levels of Testing: Traditional view of testing levels, Integration Testing (Decomposition based integration), Unit Testing, System Testing. Metrics and Complexity: Metrics definition, objectives, Linguistic Metrics: definition, LOC, Statement counts, related metrics, Halstead's metrics, token count. Structural Metrics -Definition, Cyclomatic complexity (its calculation & uses), Hybrid Metrics.

Text Books:

1. Boris Beizer, "Software Testing Techniques", dreamtech press.
2. Anirban Basu "Software Quality Assurance Testing & Metrics", PHI Learning.
3. Paul C. Jorgensen, "Software Testing: A Craftsman's approach", CRC Press.

Reference Books:

1. Alan C. Gillies, "Software Quality: Theory & Management", Cengage Learning.
2. Jeff Tian, "Software Quality Engineering", Wiley Publication.
3. Nina S Godbole, "Software Quality Assurance: Principles & Practice", Narosa Publishing House. 4. Aditya P. Mathur, "Fundamentals of Software Testing", Pearson Education

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand the basics Software Quality Assurance and use the SQA techniques efficiently.
CO2: Understand the importance of Software Quality metrics and acquire the knowledge about McCall's, Boehm, ISO 9126 and Dromey's Quality model.
CO3: Acquire the knowledge about Software testing and skill to perform functional, equivalence, and acceptance testing.
CO4: Calculate Cyclomatic Complexity and perform Unit & System testing.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-33(ii): Automata Theory and Compiler Design **L:4 T:0 P:0**

Course Objectives: Understanding the inherent capabilities and limitations of computers is a fundamental question in computer science. To answer this question, we will define formal mathematical models of computation, and study their relationships with formal languages. Topics will consist of three central areas of the theory of computation: automata, computability, and complexity. Students will also learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms. Throughout this course, they will strengthen their rigorous mathematical reasoning skills.

UNIT-I

Automata: Introduction to finite automata, structural representations, automata and complexity, Alphabets, strings, languages, problems, Chomsky hierarchy, Deterministic Finite Automata (DFA), Mealy Machine, Moore Machine, Non-deterministic Finite Automata (NFA), Finite Automata With Epsilon Transition. Convernation from NFA to DFA.

UNIT-II

Regular Expression and languages: regular expressions, finite automata and regular expressions, applications, algebraic laws, pumping lemma for regular languages, closure properties, equivalence and minimization of automata.
Context Free Grammars and languages: Introduction to context free grammars, parse trees, applications of CFG, ambiguity in grammars and languages.

UNIT-III

Pushdown Automata: definition of PushDown Automata (PDA), languages of a PDA, equivalence of PDA and CFG, deterministic PDA, non-deterministic PDA, properties of context free languages, normal forms, pumping lemma, closure properties, decision properties.

UNIT-IV

Turing Machine: Problems that computer cannot solve, the turing machine, programming techniques for Turing machines, extensions to the basic turing machine, restricted Turing machines, Turing machines and computers. Basic understanding of the classes P and NP.

Text Books:

1. J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, languages, and computation (2nd ed.), Addison-Wesley, 2001.

Reference Books:

1. H. R. Lewis, C.H. Papadimitriou, C. Papadimitriou, Elements of the Theory of Computation (2nd ed.), Prentice-Hall, NJ, 1997.

2. J. A. Anderson, Automata Theory with Modern Applications, Cambridge University Press, 2006.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand key notions of computation, such as algorithm, computability, decidability, reducibility, and complexity, through problem solving.
CO2: Explain the models of computation, including formal languages, grammars and automata, and their connections.

CO3: State and explain the Church-Turing thesis and its significance.

CO4: Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.

CO5: Solve computational problems regarding their computability and complexity and prove the basic results of the theory of computation.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.

MT-CS-25-33(iii): Graph Theory **L:4 T:0 P:0**

Course Objectives: This course offers a comprehensive exploration of graph theory, a fundamental area of mathematics with diverse applications in computer science, engineering, and various other fields. Students will learn about the basic concepts of graphs, spanning trees, planar graphs, directed graphs, and various algorithms used for solving graph-related problems.

UNIT-I

Introduction: Definition of a graph, application of graphs, finite and infinite graphs, incidence and degree, isolated vertex, pendant graph, null graph. Path and circuits- Isomorphism, subgraphs, walks, paths, circuits, connected graphs, disconnected graphs and its componenets, Euler graph, operations on graphs, Hamiltonian paths and circuits, travelling salesman problem.

UNIT-II

Trees and fundamental circuits- Trees, properties of the trees, pendant vertices in a tree, distance and centres in a tree , rooted and binary trees, on counting trees, spanning tree, fundamental circuits, finding all spanning trees of a graph, spanning tree in a weighted graph.

UNIT-III

Planar and Dual graphs: Combinatorial Vs. Geometric Graphs, planar graphs, diffirent representations of a planar graph, detection of planarity, Geometric Dual, combinatorial dual, thickness and crossings, Matrix representation of graphs- Incidence graph, submatrices of A(G), circuit matrix, cut-set matrix, path matrix adjacency matrix.

UNIT-IV

Directed Graphs: Definition of a directed graph, types of digraphs, digraphs and binary relations, directed path and connectedness, trees with directed edges, fundamental circuits in a digraph, adjacency matrix of a graph, acyclic digraphs and decyclization.

Graph Algorithms: algorithm for connectedness, a spanning tree, a set of fundamental circuits, directed circuits, shortest path algorithm, depth search first on a graph, algorithm for planarity testing, algorithm for isomorphism.

Text Books:
1. Narsingh Deo, "Graph Theory", Prentice Hall of India.

Course Outcomes:

By the end of this course, the student will be able to:

CO1: Understand and explore the basics of graph theory.
CO2: Analyse the significance of graph theory in different engineering disciplines.
CO3: Demonstrate algorithms used in interdisciplinary engineering domains.
CO4: Evaluate or synthesize any real world applications using graph theory.

Note: In each theory paper, nine questions are to be set. Two questions are to be set from each Unit and candidate is required to attempt one question from each unit. Question number nine will be compulsory, which will be of short answer type with 5-10 parts, out of the entire syllabus. In all, five questions are to be attempted.