K. S. INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU, Approved by AICTE

Department of Master of Computer Applications

SECOND SEMESTER SYLLABUS

Course : Data Structure and Algorithms		Semester	II
Course Code	**25MMC203**	CIE Marks	50
Teaching Hours/Week (L:P:T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	04	Exam Hours	03
Examination type (SEE)	**Theory**		

Course Objectives (Course Skill Set)

1. To understand and implement fundamental data structures.
2. To develop efficient algorithms for solving problems.
3. To analyze the time and space complexity of algorithms.
4. To gain practical experience in applying data structures and algorithms to real-world problems.
5. To prepare students for industry roles requiring strong foundations in data structures and algorithmic thinking

Module-1

Introduction to Data Structures and Algorithms:

Basic Concepts: Definition and importance of data structures, Abstract Data Types (ADTs), Algorithm analysis: Time and space complexity, **Big O notation. Arrays:** Definition and operations: Insertion, deletion, traversal, Multidimensional arrays, Applications of arrays.

Linked Lists: Singly linked list: Creation, insertion, deletion, traversal, Doubly linked list and circular linked list, Applications of linked lists.

Number of Hours:10

Module-2

Stacks, Queues, and Recursion:

Stacks: Definition and operations: Push, pop, peek, Applications: Expression evaluation, backtracking, function calls. **Queues:** Definition and operations: Enqueue, dequeue, front, rear, Types: Circular queue, priority queue, double-ended queue (deque), Applications of queues. **Recursion:** Definition and principles of recursion, Recursive algorithms: Factorial, Fibonacci series, Tower of Hanoi, Analysis of recursive algorithms.

Number of Hours:10

Module-3

Trees and Graphs:

Trees: Definition and terminology: Root, leaf, internal node, height, depth, Binary trees: Traversal (preorder, inorder, postorder), creation, insertion, deletion, Binary search trees (BST), AVL trees, B-trees.

Graphs: Definition and terminology: Vertices, edges, adjacency list, adjacency matrix, Graph traversal algorithms: Depth-first search (DFS), breadth-first search (BFS), Shortest path algorithms, Floyd -Warshall algorithm. **Number of Hours:10**

Module-4

Sorting and Searching Algorithms:

Sorting Algorithms: Basic concepts and classification, Comparison-based sorting: Bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort, Non-comparison-based sorting: Radix sort, counting sort. **Searching Algorithms:** Linear search and binary search, Search in linked lists, trees, and graphs, Hashing: Hash functions, collision resolution techniques (chaining, open addressing). **Number of Hours:10**

Module-5

Advanced Data Structures and Applications:

Advanced Data Structures: Heaps: Definition, operations, heap sort, applications, Trie: Definition, operations, applications in dictionary and spell-checking, Segment trees and Fenwick trees: Definition, operations, range queries. **Algorithm Design Techniques:** Divide and conquer, Greedy algorithms, Dynamic programming. **Industry Applications:** Real- world applications of data structures and algorithms, Best practices in data structure and algorithm implementation, Case studies of complex problem-solving using advanced data structures.

Number of Hours:10

Course outcome (Course Skill Set)

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

CO1: Apply fundamental data structures to develop efficient algorithms.

CO2: Apply linear data structures to solve a given problem.

CO3: Apply Non- linear data structures to solve a given problem.

CO4: Develop algorithm for sorting and searching problems.

CO5: Apply advanced data structures and algorithms to solve complex problems.

Suggested Learning Resources:

Books (Name of the author/Title of the Book/Name of the publisher/Edition and Year) Text Books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
3. Algorithms, Kenneth A Berman and Jerome L Paul, Cengage Learning India Pvt Ltd, 2002 edition.

Reference books:

- Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
- Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
- Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2 nd Ed, McGraw Hill, 2013
- A M Tenenbaum, Data Structures using C, PHI, 1989
- Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.
- Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson.
- Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=BBpAmxU_NQo
- https://www.youtube.com/watch?v=8hly31xKli0
- https://archive.nptel.ac.in/courses/106/106/106106127/

Teaching-Learning Process (Innovative Delivery Methods)

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching- learning process and facilitate the achievement of course outcomes.

1. Lectures with PowerPoint presentations, Hands-on coding exercises in C, Interactive discussions and problem-solving sessions, Assignments and quizzes for assessment.
2. Case studies and real-world examples, Practical coding sessions, Group discussions and problem-solving exercises, Mid-term project focusing on stack and queue applications.
3. Lab exercises on tree and graph implementations, Interactive coding sessions with real-time problem- solving, Group projects to develop tree and graph applications, Continuous assessment through quizzes and coding challenges.
4. Demonstrations and hands-on coding practice, Problem-solving sessions with practical use cases, Case studies on the application of sorting and searching algorithms, Assignments and group activities to solidify understanding.
5. Practical sessions on advanced data structures, Industry guest lectures, Project-based learning with real-world applications, Final project presentation and assessment.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 50% of the maximum marks. Minimum passing marks in SEE is 40% of the maximum marks of SEE. A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 50% (50 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

1. Three Unit Tests each of 25 Marks (scale down to 25 Marks)
2. Two assignments each of 25 Marks or one Skill Development Activity of 50 marks to attain the COs and POs which will be scale down to 25 marks.

The sum of **three**-unit tests, two assignments/Skill Development Activities (CIE), will be 50 marks.

Semester-End Examination:

1. The SEE question paper will be set for 100 marks and the marks scored will be proportionately reduced to 50.
2. The question paper consists of Part A and Part B. Part A consists of 10 questions from 5 modules, each carrying 2 marks.
3. Part B consists of 10 questions. Each full question is for 16 marks. There will be two full questions (with a maximum of three sub-questions) from each module.
4. Each full question will have a sub-question covering all the topics under a module.
5. The students will have to answer five full questions, selecting one full question from each module