INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

ADVANCED DATA STRUCTURES AND ALGORITHMS								
V Semester: IT								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
AITD06	**Elective**	3	-	-	3	40	60	100
Contact Classes: 48	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 48**		
Prerequisite:								

I. COURSE OVERVIEW:

This course provides an in-depth understanding of advanced data structures and algorithms, emphasizing their role in efficient problem-solving and software development. Students will explore the evolution of algorithmic techniques and their applications in real-world scenarios, alongside the development of modern programming tools and data structures. The course focuses on analyzing the trade-offs in algorithm design, understanding the principles of efficient data organization, and evaluating the success and limitations of algorithms in computational problem-solving. Students will also develop the ability to integrate theoretical concepts with practical implementations, where innovation in data structures intersects with the needs of computational efficiency and scalability.

II. COURSE OBJECTIVES:

The students will try to learn

I. The principles of algorithm design and evaluate performance using time and space complexity.
II. The linear and nonlinear data structures in solving real-time problems.
III. The advanced tree structures, hashing techniques, and graph algorithms, along with their applications in modern computing.

III. COURSE OUTCOMES:

At the end of the course, students should be able to:

CO 1. Analyze the time and space complexity of algorithms to evaluate their efficiency and scalability.
CO 2. Design and implement advanced data structures such as AVL trees, B-trees, and hash tables to optimize data storage and retrieval.
CO 3. Apply algorithmic techniques such as dynamic programming, divide and conquer, and greedy algorithms to solve computational problems.
CO 4. Evaluate the performance of various sorting, searching, and graph algorithms to recommend the most suitable approach for a specific application.
CO 5. Implement graph traversal techniques like DFS and BFS and apply them to real-world problems such as shortest path and spanning tree computation.
CO 6. Create innovative solutions for complex computational challenges by integrating advanced algorithms and data structures in practical applications.

IV. COURSE CONTENT:

MODULE – I: INTRODUCTION TO ALGORITHMS AND LINEAR DATA STRUCTURES (09)
Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Data structures-Linear and nonlinear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

MODULE – II: STACKS, QUEUES, AND PRIORITY QUEUES (08)
Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-Array List, Linked List, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

MODULE – III: SEARCHING, HASHING, AND SORTING TECHNIQUES (10)
Searching–Linear and binary search methods, Hashing-Hash functions, Collision Resolution Methods-Open Addressing, Chaining, Hashing in java. util-HashMap, HashSet, Hash table.

Sorting –Bubble sort, Insertion sort, Quick sort, Merge sort, Heap sort, Radix sort, comparison of sorting methods.

MODULE - IV TREES, GRAPHS, AND THEIR APPLICATIONS (10)
Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non-recursive traversals, Java code for traversals, Threaded binary trees. Graphs- Graphs terminology, Graph ADT, representations, graph traversals/search methods-dfs and bfs, Java code for graph traversals, Applications of Graphs-Minimum cost spanning tree using Kruskal's algorithm, Dijkstra's algorithm for Single Source Shortest Path Problem.

MODULE –V: ADVANCED SEARCH TREES AND TEXT PROCESSING ALGORITHMS (08)
Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition and examples only, Red Black trees – Definition and examples only, B-Trees-definition, insertion and searching operations, Trees in java.util- TreeSet, Tree Map Classes, Tries(examples only), Comparison of Search trees. Text compression-Huffman coding and decoding, Pattern matching-KMP algorithm.

V. TEXT BOOKS:

1. S. Sahni, "Data structures, Algorithms and Applications in Java", Universities Press. [ISBN:0-07-109217-x]
2. Denis Adam Drozdek, "Data structures and Algorithms in Java", 3rd edition, Cengage Learning. [ISBN:978-9814239233].

VI. REFERENCE BOOKS:

1. Philip Lesley, R.Lafore "Data structures and Algorithms in Java", Pearson education. ISBN: 9788 131718124
2. J.P.Tremblay and G.A.Cheston "Data structures and Software Development in an Object-Oriented Domain", Java edition, Pearson Education.

VII. WEB REFERENCES:

1. https://www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm
2. https://www.codechef.com/certification/data-structures-and-algorithms/prepare

VIII. MATERIALS ONLINE

1. Course template
2. Tutorial question bank
3. Tech talk topics

4. Open end experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper - II
9. Lecture notes
10. E-learning readiness videos (ELRV)
11. Power point presentation