

24AI3601	DATASTRUCTURES AND ALGORITHM ANALYSIS  (Common to AI-DS, CSBS and AIML)	L	T	P	C
		3	0	0	3
Preamble					
Data structures course focusing on effective programming than the syntax / semantics of any programming language. Algorithms will be discussed greedy algorithms, advanced data structures, graph algorithms NP completeness.					
Prerequisites for the course					
<ul style="list-style-type: none"><li>24CS1501-Introduction to programming with C</li></ul>					
Objectives					
<ol style="list-style-type: none"><li>To understand the concepts of ADTs linear data structures – lists, stacks, and queues</li><li>To apply non-linear data structure operations</li><li>To understand hashing algorithms and sorting applications.</li><li>To learn information about algorithms, asymptotic notations, and performance analysis algorithms.</li><li>To Develop dynamic programming algorithms for various real-time applications. Illustrate and apply backtracking algorithms, further able to understand non-deterministic algorithms.</li></ol>					
UNIT I	LINEAR AND NON LINEAR DATA STRUCTURES	9			
Abstract Data Types (ADT) - List ADT – linked list implementation – cursor-based linked lists – doubly-linked lists – applications of lists – Stack ADT Operations – Applications: Evaluating arithmetic expressions- Conversion of Infix to postfix expression – Queue ADT – circular queue implementation – Applications of stacks and queues					
SUGGESTED ACTIVITIES:					
<ul style="list-style-type: none"><li>Practice designing algorithms for some small simple problems, proving their correctness, and estimate their complexity</li><li>Converting an algorithm from recursive to non-recursive using stack</li></ul>					
SUGGESTED EVALUATION METHODS:					
<ul style="list-style-type: none"><li>Assignment-Based on design, correctness and efficiency</li><li>Quizzes</li></ul>					
UNIT II	TREE AND GRAPH STRUCTURES	9			
Tree ADT - Binary Tree ADT - binary search tree - AVL trees - binary heaps - Representation of Graph – Types of graph - Breadth-first traversal - Depth-first traversal – Minimum Spanning Trees – Kruskal and Prim algorithm – Shortest path algorithm – Dijkstra’s Algorithm					
SUGGESTED ACTIVITIES:					
<ul style="list-style-type: none"><li>Applications of trees.</li><li>Practical Implementing graph traversals.</li></ul>					



**SUGGESTEDEVALUATIONMETHODS:**

- Assignment Problem
- Quizzes

**UNIT III****HASHING AND SORTING****9**

Hashing – Separate chaining - open addressing – rehashing - extendible hashing - Sorting - Bubble sort - Selection Sort, Insertion Sort- Quick sort ,Merge sort- Hashing - Hash Functions

**SUGGESTEDACTIVITIES:**

- Practical–Implementation of Hash table

**SUGGESTEDEVALUATIONMETHODS:**

- Assignment related to application
- Programming exercises in the laboratory
- Quizzes

**UNITIV****INTRODUCTION TO ALGORITHM****9**

Introduction to algorithm analysis: Algorithm Specification - Analysis Framework - **Performance Analysis:** Space complexity, Time complexity - Asymptotic Notations: Big-Oh notation (O), Omega notation ( $\Omega$ ), Theta notation.

**SUGGESTEDACTIVITIES:**

- Comparison of Asymptotic Notation

**SUGGESTEDEVALUATIONMETHODS:**

- Assignment Problem
- Programming exercises in the laboratory
- Quizzes

**UNITV****ALGORITHM DESIGN AND ANALYSIS****9**

Introduction to algorithm design techniques: Greedy algorithms, Divide and conquer: General method, Binary search, Recurrence equation for divide and conquer, Dynamic programming: Knapsack problem ,Bellman-Ford Algorithm, backtracking: N-Queens problem, branch and bound, Randomized algorithms

**SUGGESTEDACTIVITIES:**

- Comparison of algorithms Analysis