

## DATA STRUCTURES

IV Semester AE / EEE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSB03	Core	L	T	P	C	CIA	SEE	Total
		3	0	0	3	30	70	100
<b>Contact Classes: 45</b>		<b>Tutorial Classes: Nil</b>		<b>Practical Classes: Nil</b>		<b>Total Classes: 60</b>		
<p><b>OBJECTIVES:</b>  <b>The students will try to learn:</b></p> <p>I To provide students with skills needed to understand and analyze performance trade-offs of different algorithms / implementations and asymptotic analysis of their running time and memory usage.</p> <p>II To provide knowledge of basic abstract data types (ADT) and associated algorithms: stacks, queues, lists, tree, graphs, hashing and sorting, selection and searching.</p> <p>III The fundamentals of how to store, retrieve, and process data efficiently.</p> <p>IV To provide practice by specifying and implementing these data structures and algorithms in Python.</p> <p>V Understand essential for future programming and software engineering courses.</p> <p><b>COURSE OUTCOMES:</b>  <b>After successful completion of the course, students will be able to:</b></p> <p>CO 1 Carryout the analysis of a range of algorithms in terms of algorithm analysis and express algorithm complexity using the O notation.</p> <p>CO 2 Make use of recursive algorithm design technique in appropriate contexts.</p> <p>CO 3 Represent standard ADTs by means of appropriate data structures.</p> <p>CO 4 Select appropriate sorting technique for given problem.</p> <p>CO 5 Select appropriate searching technique for given problem.</p> <p>CO 6 Implement standard searching and sorting algorithms; including binary search; merge sort and quick sort; and analyze their time and space complexities.</p> <p>CO 7 Implement linked lists, stacks and queues in Python for problem solving.</p> <p>CO 8 Explain the use of basic data structures such as arrays, stacks, queues and linked lists in program design.</p> <p>CO 9 Extend their knowledge of data structures to more sophisticated data structures to solve problems involving balanced binary search trees, AVL Trees, B-trees and B+ trees, hashing, and basic graphs.</p> <p>CO 10 Design and implement tree structures in real-time applications.</p> <p>CO 11 Compare and contrast the benefits of dynamic and static data structures implementations and choose appropriate data structure for specified problem domain.</p> <p>CO 12 Determine and explain how efficient an algorithm or data structure will be, apply appropriate data structures for solving computing problems with respect to performance.</p>								

<b>MODULE-I</b>	<b>INTRODUCTION TO DATA STRUCTURES, SEARCHING AND SORTING</b>
Basic concepts: Introduction to data structures, classification of data structures, operations on data structures; Algorithm Specification, Recursive algorithms, Data Abstraction, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega, and Theta notations. Introduction to Linear and Non Linear data structures, Searching techniques: Linear and Binary search; Sorting techniques: Bubble, Selection, Insertion, Quick and Heap Sort and comparison of sorting algorithms.	
<b>MODULE-II</b>	<b>LINEAR DATA STRUCTURES</b>
Stacks: Stack ADT, definition and operations, Implementations of stacks using array, applications of stacks, Arithmetic expression conversion and evaluation; Queues: Primitive operations; Implementation of queues using Arrays, applications of linear queue, circular queue and double ended queue (deque).	
<b>MODULE-III</b>	<b>LINKED LISTS</b>
Linked lists: Introduction, singly linked list, representation of a linked list in memory, operations on a single linked list; Applications of linked lists: Polynomial representation and sparse matrix manipulation.  Types of linked lists: Circular linked lists, doubly linked lists; Linked list representation and operations of Stack, linked list representation and operations of queue.	
<b>MODULE-IV</b>	<b>NON LINEAR DATA STRUCTURES</b>
Trees: Basic concept, binary tree, binary tree representation, array and linked representations, binary tree traversal, binary tree variants, threaded binary trees, application of trees, Graphs: Basic concept, graph terminology, Graph Representations - Adjacency matrix, Adjacency lists, graph implementation, Graph traversals – BFS, DFS, Application of graphs, Minimum spanning trees – Prims and Kruskal algorithms.	
<b>MODULE-V</b>	<b>BINARY TREES AND HASHING</b>
Binary search trees: Binary search trees, properties and operations; Balanced search trees: AVL trees; Introduction to M-Way search trees, B trees; Hashing and collision: Introduction, hash tables, hash functions, collisions, applications of hashing.	
<b>Text Books:</b>	
1. Rance D. Necaise, “Data Structures and Algorithms using Python”, Wiley Student Edition. 2. Benjamin Baka, David Julian, “Python Data Structures and Algorithms”, Packt Publishers, 2017.	
<b>Reference Books:</b>	
1. S. Lipschutz, “Data Structures”, Tata McGraw Hill Education, 1 <sup>st</sup> Edition, 2008. 2. D. Samanta, “Classic Data Structures”, PHI Learning, 2 <sup>nd</sup> Edition, 2004.	
<b>Web References:</b>	
1. <a href="https://www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm">https://www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm</a> 2. <a href="https://www.codechef.com/certification/data-structures-and-algorithms/prepare">https://www.codechef.com/certification/data-structures-and-algorithms/prepare</a> 3. <a href="https://www.cs.auckland.ac.nz/software/AlgAnim/dsToC.html">https://www.cs.auckland.ac.nz/software/AlgAnim/dsToC.html</a> 4. <a href="https://online-learning.harvard.edu/course/data-structures-and-algorithms">https://online-learning.harvard.edu/course/data-structures-and-algorithms</a>	