

Year: B. Tech II (Semester III)

Subject Name: Data Structures

Subject Code: BTIT13302

Type of course: Professional Core Course

Prerequisite (if any): Basic Programming Skills (Python, C, C++, JAVA)

Rationale: Data Structures is a subject of primary importance in Information Technology. Organizing or structuring data is important for efficient implementation of algorithms and program development. Efficient problem solving needs the application of appropriate data structures during program development.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	2	4	60	25	15	30	20	150

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests) CA2: Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Contents	Total Hours
1	Introduction To Data Structures: Data Types – Primitive and Non-primitive, Types of Data Structures - Linear & Non Linear Data Structures.	03
2	Linear Data Structures: Array: Representation of arrays, Applications of arrays, sparse matrix and its representation Stack: Stack-Definitions & Concepts, Operations on Stacks, Applications of Stacks, Polish Expression, Reverse Polish Expression and their compilation, Recursion, Tower of Hanoi Queue: Representation of Queue, Operations on Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue Linked List: Singly Linked List, Doubly Linked list, Circular Linked list, Linked implementation of Stack, Linked implementation of Queue, Applications of Linked List.	15

3	Non-linear Data Structures: Tree-Definitions and Concepts, Representation of Binary tree, Binary tree traversal (Inorder, postorder, preorder), Threaded Binary Tree, Binary Search Trees, Conversion of General Trees To Binary Trees, Applications Of Trees- Some balanced tree mechanism, E.g. AVL trees, 2-3 trees, Height Balanced, Weight Balanced, Graph-Matrix Representation of Graphs, Elementary Graph operations (Breadth First Search, Depth First Search, Spanning Trees, Shortest path, Minimal Spanning tree)	15
4	Hashing: The symbol table, Hashing Functions, Collision- Resolution Techniques.	06
5	Sorting & Searching: Sorting – Bubble Sort, Selection Sort, Quick Sort, Merge Sort, Searching – Sequential Search and Binary Search	06

Suggested Specification table with Marks (Theory): (For B.Tech only)

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15	20	20	5	0	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr. No	Title of Book /Article	Author(s)	Publisher and like ISBN	Year of Publication	Publication Edition
1	An Introduction to Data Structures with Applications	Paul Tremblay Paul Sorenson	McGraw Hill Education	2017	2 nd Edition
2	Data structures using C	E. Balagurusamy	McGraw Hill	2017	1 st Edition

Course Outcomes:

Sr. No.	CO statements	Marks % weightage
CO-1	Understand and explain Primitive and Non Primitive data structures	10%
CO-2	Design and apply appropriate data structures for solving computing problems	10%
CO-3	Describe and implement linear data structures	25%
CO-4	Describe and implement non-linear data structures	25%
CO-5	Implement sorting and searching techniques	20%
CO-6	Understand and explain hash functions and collision resolution techniques	10%

For Lab Sessions:

List of Experiments:

Sr. No Practical Statements

1. Implement a program to demonstrate Call by Value and Call by reference.
2. Implement a program to create array using Dynamic Memory Allocation.
3. Implement a program for stack that performs following operations using array.
(a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
4. Implement a program to convert infix notation to postfix notation using stack.
5. Write a program to implement QUEUE using arrays that performs following operations
(a) INSERT (b) DELETE (c) DISPLAY
6. Write a program to implement Circular Queue using arrays that performs following operations.
(a) INSERT (b) DELETE (c) DISPLAY
7. Write a menu driven program to implement following operations on the singly linked list.
(a) Insert a node at the front of the linked list.
(b) Insert a node at the end of the linked list.
(c) Insert a node such that linked list is in ascending order.(according to info. Field)
(d) Delete a first node of the linked list.

- (e) Delete a node before specified position.
 - (f) Delete a node after specified position.
8. Write a program to implement stack using linked list.
 9. Write a program to implement queue using linked list.
 10. Write a program to implement following operations on the doubly linked list.
 - (a) Insert a node at the front of the linked list.
 - (b) Insert a node at the end of the linked list.
 - (c) Delete a last node of the linked list.
 - (d) Delete a node before specified position.
 11. Write a program to implement following operations on the circular linked list.
 - (a) Insert a node at the end of the linked list.
 - (b) Insert a node before specified position.
 - (c) Delete a first node of the linked list.
 - (d) Delete a node after specified position.
 12. Write a program which create binary search tree.
 13. Implement recursive and non-recursive tree traversing methods inorder, preorder and post-order traversal.
 14. Write a program to implement Quick Sort.
 15. Write a program to implement Merge Sort.
 16. Write a program to implement Bubble Sort.
 17. Write a program to implement Binary Search.

List of Open learning website:

- <https://www.programiz.com/dsa>
- https://www.tutorialspoint.com/data_structures_algorithms/index.htm
- <https://www.javatpoint.com/data-structure-tutorial>

List of Open Source Software:

- Code: Blocks
- CodeLite