

Data Structures & Algorithms-Lab					
Credit Hours:	1	Course Code:	INFT-2207	Prerequisites:	None
Course Learning Outcomes (CLOs):					
At the end of the course, the students will be able to:				Domain	BT Level*
1. Implement various data structures and their algorithms and apply them in implementing simple applications				C	1
2. Apply the knowledge of data structure to other application domains.				C	2
3. Analyze simple algorithms and determine their complexities.				C	3
4. Design new data structures and algorithms to solve problems.				C	3
* BT= Bloom’s Taxonomy, C=Cognitive Domain, P=Psychomotor Domain, A=Affective Domain					
#	SDG	Relevant CLOs			
1	SDG 4: Quality Education	CLO 2: Analyze simple algorithms and determine their complexities.			
2	SDG 9: Industry, Innovation and Infrastructure	CLO 3: Apply the knowledge of data structure to other application domains. CLO 4: Design new data structures and algorithms to solve problems.			
Course Contents:					
Arrays, Functions and Pointers; Array implementation; Array and functions; Pointers; Complexity Analysis; Asymptotic notations; Types of complexities (Best, Average, Worst cases); Big oh, big omega, big theta; Complexity analysis of code snippets; Searching & Sorting in Arrays; Linear search; Binary search; Selection sort; Insertion sort; Bubble sort; Quick sort; Merge sort; Comparison using asymptotic notations; Linked Lists, Types of Linked Lists (Singly, Doubly, Circular); Algorithms (Searching, sorting, insertion, deletion); Stacks: PUSH, POP, PEEK functions; Implementation using Arrays; Implementation using Linked Lists; Expression parsing using Stacks (infix, prefix, postfix); Queues: Queue functions (Enqueue, Dequeue, search, Sort); Implementation using Arrays; Implementation using Linked Lists; Implementation of different types of queues (circular buffer, double ended, priority); Hashing; Recursion exercises: Fibonacci series, Tower of Hanoi puzzle; Tree: Tree traversals (inorder, preorder, postorder); Binary Search Trees: insertion of node, deletion of node, traversals, searching a node; Expression trees; Heap: Min heap, Max heap; Graphs: Searching in graph (BFS, DFS), Topological order, shortest path first; Dynamic programming exercises; Greedy algorithm exercises; Backtracking algorithm exercises; Amortized analysis exercises; Minimum Spanning Tree implementations; Algorithm correctness and debugging; Performance benchmarking and testing algorithms on large inputs.					
Teaching Methodology:					
Lectures, Written Assignments, Practical labs, Semester Project, Presentations.					
Course Assessment:					
Mid Term Exams, Final Term Exams, Assignments, Quizzes, Project, Presentations, Practical Exam					
Reference Material:					
1. Data Structures and Algorithms in Java: A Project-Based Approach by Dan S. Myers, Year 2024, ISBN-13: 978-1009260336 2. Data Structure and Algorithms in C++ by Adam Drozdek, 4th Edition, ISBN-13: 978-1133608424 3. Data Structure and Algorithms in java by Adam Drozdek, 4th Edition, ISBN-13: 978-9814392785 4. Data Structures and Algorithm Analysis in Java by Mark A. Weiss, 3rd Edition, ISBN-13: 978-0132576277 5. Java Software Solutions: foundation of Program Design by John Lewis and Joseph Chase, 9th Edition, ISBN-13: 978-1292221724					