Title	**Data Structures Lab**		
Code	**CC-213-L**		
Credit Hours	1 (0,3)		
Category	Computing Core		
Prerequisite	CC-211 Object Oriented Programming		
Co-Requisite	None		
Follow-up			
Course Introduction	The course is designed to teach students structures and schemes, which allow them to write programmer to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of time and space complexity of computer programs.		
Course Learning Outcomes (CLOs)	At the end of the lab, the students will be able to:	**BT**	**PLO**
	CLO1: Implement various data structures and their algorithms and apply them in implementing simple applications	C3 (Apply)	1,2
	CLO2: Analyze simple algorithms and determine their complexities.	C4 (Analyze)	3
	CLO3: Apply the knowledge of data structure to other application domains	C3 (Apply)	3,4
	CLO4: Design new data structures and algorithms to solve problems	C6 (Design)	4,5
Course Description	***Implementation and Practice of the concepts studied in "CC-213 Data Structures"*** **Algorithm Specification:** Properties of Algorithm, examples, performance, analysis, measurement, and Big Oh notation. **Introduction to ADTs:** Array and Polynomial as an ADT, Sparse Matrices, and Representation of Arrays. **The Stack ADT:** Linked list and array implementations, Expressions, Postfix Notation, and Infix to postfix conversion. **The Queue ADT:** Linked and array implementations of circular and double ended queue. **Recursion:** Recursive Definition and Processes, Writing Recursive Programs. Divide and Conquer Algorithms, Self-Referencing Classes and Dynamic Memory Allocation, Garbage Collection. **Linked List:** Singly Linked Lists, Circular Lists, Linked Stacks and Queues (Double Ended List), Doubly Linked Lists. **Trees:** Introduction to Trees, Logical construction and Traversing of Binary Trees, Implementation of Binary Trees (Insertion and Traversing), Searching and deletion in Binary Trees, Binary Search Tree, Introduction to Balanced and AVL Trees. **Heaps:** Heaps and Heaps as Priority Queues, Double Ended Priority Queue. **Searching:** Linear Search, Binary Search, and Types of Indexing. **Hashing:** Hash Functions: Division, Open Addressing; Overflow Handling: Chaining; **Introduction to advanced topics:** B-Trees, M-Way Trees, Generalized List etc. **Sorting:** Selection, Insertion, Merge, Quick, Bubble, Heap, Shell, Radix, and Bucket sorts. **Graphs:** Graph terminology, Adjacency List and Adjacency Matrix and Adjacency list representation of Graph; Elementary Graph Operations: Breadth First Search and Depth First Search, Spanning Trees (BFSST, DFSST), topological order, shortest path.		
Text Book(s)	A. Ellis Horowitz, Sartaj Sahni, and D. Mehta, "Fundamentals of Data Structures in C++", 2nd Ed., Computer Science Press		

Reference Material	1. Adam B. Drozdek, Data Structure and Algorithm in C++, 4th Ed., Cengage Learning 2. Mark Allen Weiss, "Data Structure and Algorithms in C++", 2nd Ed., Pearson Education 3. D. Malhotra and N. Malhotra. Data Structures and Program Design Using C++. 4. Tenenbaum, M. Augenstein, and Y. Lang Sam, "Data Structures using C and C++" 2nd Ed., Prentice Hall