GCWUS-Sialkot

Title	Data Structures and Algorithm
Code	
Credit Hours	Weight 4 Cr. Hrs Lecture 1 Lecture / Week
Lecture Days	Monday (12:30PM to 3:30 PM)
Instructor	**Dr. Usman Ashraf**
Prerequisite	**Programming fundamental**
Aims and Objectives	Algorithms and data structures emphasizes the following topics: data structures, abstract data types, recursive algorithms, algorithm analysis, sorting and searching, and problem-solving strategies.
Learning Outcomes	● Understand numerous examples of relationships between data. ● Understand the abstract data types of stacks, queues and deques; ● Understand the variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated; ● Understand the characteristics and optimal behavior of hash tables for access and retrieve. ● Understand various graph algorithms.
Syllabus	Introduction to Data Structures and Algorithms; Complexity Analysis; Arrays; Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort, Bucket Sort; Linked Lists: Singly Linked Lists, Doubly Linked Lists, Circular List; Stacks, Queues, and Priority Queue; Recursion: Function call and Recursion Implementation, Tail Recursion, Non-tail Recursion, Indirect Recursion, Nested Recursion, Backtracking. Trees: Binary Trees, Binary Heap, Binary Search. Tree Traversal, Insertion, Deletion, and Balancing a Tree; Heap; B-Tree; Spanning Tree, Splay Trees; Graphs: Representation, Traversal, Shortest Path, and Cycle Detection; Isomorphic Graphs; Graph Traversal Algorithms; Hashing; Memory Management and Garbage Collection.
Text Book/s	● Mark Allen Weiss, “Data Structure and Algorithms in C++”, Benjamin/Cumming
Reference Material	● D. Samanta. “Classic Data Structures”, Prentice Hall, 2001 ● Handouts ● *Algorithms*, Robert Sedgewick, Princeton University Publisher: Addison- Wesley Professional (latest Edition)

Assessment Criteria				
	Sessional 25%	**Mid 35%**	**Final 40%**	**Total 100%**
	Quizzes 3 Assignment 3 Project +presentations 10	Required:	Required:	
		Paper 30	Paper 50	
Recommendations				

Weeks	**Topics description**	**Assignments/ Quizzes (Unannounced) Expected 5,5.**
1	**Course Introduction:** ● Course Policies/Overview ● Course Contents/Course Objectives ● Introduction to data structures and types	
2	**Stack** ● Introduction to stack, definitions. ● Abstract data types of stack and their implementations. ● Applications of stack. ● Evaluating an expression in postfix form, Converting infix expression to postfix	
3	**Operations on Stack:** ● Stack operation ● Implementation of stack using array. ● Implementation of stack using linked list. ● Applications of stack.	
4	**List as Simple Data Structure** ● Simple List as Array ● Operation on list ● Traversing, Searching, Inserting and Deleting data element	
5	**Linked List** ● Concept of pointer and linked list ● Static and dynamic variables ● Types of Linked list ● Implementation of linked list using pointer	
6	**Queues:** ● Introduction to queue, definitions. ● Physical model, linear implementation, circular queue. ● Abstract data types of queue and their implementations.	

	● Applications of queue.	
7	**Operations on Queue:** ● Insertion and deletion ● Traversing and searching ● Applications of queue structure	
8	**Operations on Linked List** ● Traversing and Searching data element ● Inserting and Deleting data element ● Doubly link list ● Implement a doubly-linked list	
9		
10	**Searching and Sorting Algorithms** ● Algorithms for sorting ● Insertion sort, bubble sort, merge sort, quick sort, heap sort. ● Sequential search: searching an ordered table, index sequential search. ● Compare different sorting and searching algorithms.	
	Tree ● Introduction to tree: definitions. ● Tree size, level and depth in tree. ● Binary tree. Implementation of binary tree.	
11	**Operations on Tree** ● Expression tree (Pre-order, In-order and Post-order) ● Binary Search Tree	
12	**Heaps** ● Introduction ● Max Heap ● Min Heap ● Min-Max Heap	
13	**Graphs** ● Definition of graph ● Graph traversals ● Shortest path (greedy algorithm) ● Graphs as data structures	
14	**Recursive Algorithms** ● Base and general case ● Applications such as recursive evaluation of the binomial coefficient and recursive traversal of a linked list ● Comparison of recursive and iterative algorithms	
15	**Hash Table and Time efficiency of algorithms** ● Hashing algorithms ● Big-O notation and time-behavior ● Simple examples	
16	**Presentations**	
17	**Final Term**	