Sixteen Week Plan
Faculty of Computing & Information Technology
Department of Computer Science
Hafiz Hayat Campus, University of Gujrat

	Title	**Data Structure (DS)**
	Code	**CS-204**
	Credit hours	**3**
	Course Coordinator	
	Course Description	This course introduces students to new types of data structures such as trees (including binary and multiway trees), heaps, stacks and queues. Students will also learn how to design new algorithms for each new data structure studied, create and perform simple operations on graph data structures, describe and implement common algorithms for working with advanced data structures and recognize which data structure is the best to use to solve a particular problem.
	• **Objectives**	After you have completed the work in this course, you will be able to: •Develop sound techniques on designing, developing, and documenting well-structured programs using proper software engineering principles. •Continue to apply problem solving skills and provide a foundation for advanced programming courses using an OOP (object-oriented programming) methodology. •Describe and implement common data structures--lists, stacks, queues, graphs, and trees--for solving complex programming problems. •Use mathematical techniques to analyse the efficiency of the various algorithms presented, as well as the common operations on the data structures discussed
	Grading Policy	a) course will be evaluated on the following basis's: Quizzes 05 % Assignments 10 % Mid Term Exam 25 % End Semester Exam 50 % Project 10% Class Participation (marks may be allocated if the need arises) b) To pass a course, student must obtain at least 'D' grade (50% marks) c) The final term examination will cover the entire course.

	Grading System	

Marks in Percentage	Letter Grade	Numeric Value of Grade	Description
85 and above	A+	4.00	Exceptional
80-84	A	3.70	Outstanding
75-79	B+	3.40	Excellent
70-74	B	3.00	Very Good
65-69	B-	2.50	Good
60-64	C+	2.00	Average
55-59	C	1.50	Satisfactory
50-54	D	1.00	Pass
49 and below	F	0.0	Fail
	W		Withdrawal
	I		Incomplete

	Class Attendance	A minimum of 70% attendance is required for a student to be eligible to sit in the final examination
	Text Book	1. Introduction to Algorithms, (2nd ed. 2007 MIT Press) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein.
	Reference Books	1. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss. 2. Data Structures using C++, Tenanbaum
	Pre-requisites:	OOP, Introduction to programming
	Plagiarism Policy:	Collaboration and group work is encouraged but each student is required to submit his/her own contribution(s). Your writings must be your own thoughts. Cheating and plagiarism will not be tolerated and will be referred to the HoD& Dean for appropriate action(s).
	Quiz/Assignments Policy	Quizzes will be announced/unannounced. Late submission of assignments will either not be entertained or will result in the deduction of marks.

Week#	Lecture #	TOPICS	Source (Book-Chapter No)	Recommendations for Learning Activities
01	**01, 02**	Introduction to Data structures, need for data structures, Array data structure and their operations		
02	**03, 04**	Linked List , circular Link List: concepts, operations, and applications		
03	**05,06**	doubly Link List: concepts, operations, and applications		
04	**07,08**	Stack: concepts, operations, and array implementations, stack applications		
05	**09,10**	Queue , D-Queue and Circular Queue: concepts and operations		
06	**11,12**	Recursion: definitions, concepts and applications		
07	**13,14**	Trees and Binary Trees: concepts, storage representation, and manipulation		
08	**15,16**	Binary Search Tree and AVL Trees: concepts, storage representation, and manipulation		
Mid Term Exam				
09	**17,18**	Graphs: concepts, matrix and linked implementations, and traversals		
10	**19,20**	Heap: concepts, storage representation, and manipulation and heap sort		
11	**21,22**	Hashing, radix searching		
12	**23,24**	Introduction to Algorithm, Algorithms Complexity		
13	**25,26**	Sorting Algorithms (Selection, insertion, Bubble.)		
14	**27,28**	Sorting Algorithms (Selection, insertion, Bubble.)		
15	**29,30**	Divide & Conquer (Merge Sort, Quick Sort)		
16	**31,32**	Greedy algorithms		
Final Term Exam				