

COURSE DESCRIPTION

NAME OF INSTITUTION	Lahore Garrison University
PROGRAM (S) TO BE EVALUATED	Computer Science , Spring 2022

Course Description : Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm • Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors. • Determine informally the time and space complexity of simple algorithms • List and contrast standard complexity classes • Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms • Use of the strategies(brute-force, greedy, divide-and-conquer, and dynamic programming) to solve an appropriate problem • Solve problems using graph algorithms, including singlesource and all-pairs shortest paths, and at least one minimum spanning tree algorithm • Trace and/or implement a string-matching algorithm

Course Code	CSC354								
Course Title	Design & Analysis of Algorithms								
Credit Hours	3+0								
Prerequisites by Course(s) and Topics	Course : Data Structure & Algorithms Topics : Complexity Analysis; Arrays; Sorting Algorithms; Linked Lists; Queues; Recursion; Tree Traversal; Graphs Representation & Traversal.								
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	SESSIONAL (Quizzes, Assignments, Presentations) =25 % Midterm Exam =25 % Final Exam = 50%								
Course Coordinator	Adeel Munawar								
URL (if any)									
Current Catalog Description									
Textbook (or Laboratory Manual for Laboratory Courses)	● Introduction to Algorithms, T. H. Cormen, C. E. Leiserson, and R. L.Rivest, MIT Press, McGraw-Hill, 3rd Edition, New York, NY, 2010.								
Reference Material	● Algorithms in C++; Robert Sedgewick								
Course Goals	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm • Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors. • Determine informally the time and space complexity of simple algorithms • List and contrast standard complexity classes • Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms • Use of the strategies(brute-force, greedy, divide-andconquer, and dynamic programming) to solve an appropriate problem • Solve problems using graph algorithms, including singlesource and all-pairs shortest paths, and at least one minimum spanning tree algorithm • Trace and/or implement a string-matching algorithm								
Course Learning Outcomes (CLOs):	<table><tr><td>At the end of the course the students will be able to:</td><td>Domain</td><td>BT Level*</td></tr><tr><td>* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain</td><td></td><td></td></tr></table>			At the end of the course the students will be able to:	Domain	BT Level*	* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		
At the end of the course the students will be able to:	Domain	BT Level*							
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain									

Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)	Week	Lecture	Topics Covered
	Week 1	1	Introduction to Algorithm
		2	Structured description of algorithm, Three Examples of Algorithms
	Week 2	3	Algorithm Analysis (Algorithm Correctness)
		4	Algorithm Analysis (Algorithm Efficiency)
	Week 3	5	Growth of Functions: Asymptotic notation and examples of Big Oh, Big Omega, and Theta.
		6	Sorting Problem: Insertion Sort
	Week 4	7	Analysis of Insertion Sort, Loop invariants of Insertion Sort.
		8	Divide-and-Conquer Approach: Merge Sort: Example, Complete Working
	Week 5	9	Quick Sort: Example, Complete Working
		10	Heap Sort: Heaps, Maintaining the Heap property
	Week 6	11	Heap Sort: Building a Heap, The Heap Sort algorithm Sorting in Sorting in Linear Time: Radix sort
		12	Linear Time: Bucket sort Sorting in Linear Time: Counting sort
	Week 7	13	The substitution method for solving recurrences, The master method for solving recurrences
		14	The recursion-tree method for solving recurrences
	Week 8	1 hours	Mid Term
	Week 9	15	Dynamic Programming Element of the dynamic programming
		16	Matrix Chain Multiplication
	Week 10	17	Greedy Algorithms Activity Selection Problem
		18	Element of the greedy strategy
	Week 11	19	Greedy Vs Dynamic programming Knapsack Problem
		20	Huffman Trees and Codes
	Week 12	21	Huffman Trees and Codes
		22	Elementary Graph Algorithms Representation of graph
	Week 13	23	Breadth-First Search Depth-First Search
		24	Topological Sort
	Week 14	25	Minimum Spanning Tree
		26	Growing a Minimum Spanning Tree
	Week 15	27	Kruskal's Algorithm Example
		28	Prim's Algorithm Example
	Week 16	29	Single-Source Shortest Paths
		30	Single-Source Shortest Paths in DAG Dijkstra's Algorithm
	Week 17	2 hours	Final Term
Laboratory Projects/Experiments Done in the Course			
Programming Assignments Done in the Course			

Instructor Name	Adeel Munawar
Instructor Signature	
Date	