



National Computing Education Accreditation Council NCEAC



NCEAC.DOC.001-E

COURSE DESCRIPTION

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

Course Description : This is a graduate level course. The major objective of this course is providing comprehensive knowledge of modern computer algorithms and solving scientific and engineering problems efficiently and accurately. The students will be guided, how to analyze complex algorithms comparing efficiencies of these algorithms. Students will not only be taught the design of the existing algorithms but on the other hand it will be focused to teach them designing techniques using rigorous mathematical approaches. The students will be motivated to think about procedures solving real world problems optimally and correctly. Real world problem will be taken as examples to create feelings about the usefulness of this course.

Course Code	CSC																		
Course Title	Advance Analysis of Algorithm																		
Credit Hours	1																		
Prerequisites by Course(s) and Topics																			
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	Dr. Tahir Alyas																		
URL (if any)																			
Current Catalog Description																			
Textbook (or Laboratory Manual for Laboratory Courses)	CLRS, Introduction to Algorithms (3rd Ed)																		
Reference Material	S. Dasgupta, Algorithms																		
Course Goals																			
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>Students should develop a sound theoretical understanding of advanced algorithms and practical problem solving skills using them.</td><td>BT</td><td>1</td></tr><tr><td>Analyze average and worst-case running times of given algorithm.</td><td>BT</td><td>1</td></tr><tr><td>Students should gain a good understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems</td><td>BT</td><td>2</td></tr><tr><td>Analyze and design algorithms on further advanced topics such as computational geometry, operations research, cryptography, number theoretic, algorithms etc.</td><td>BT</td><td>1</td></tr><tr><td colspan="3">* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain</td></tr></table>	At the end of the course the students will be able to:	Domain	BT Level*	Students should develop a sound theoretical understanding of advanced algorithms and practical problem solving skills using them.	BT	1	Analyze average and worst-case running times of given algorithm.	BT	1	Students should gain a good understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems	BT	2	Analyze and design algorithms on further advanced topics such as computational geometry, operations research, cryptography, number theoretic, algorithms etc.	BT	1	* 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*																	
Students should develop a sound theoretical understanding of advanced algorithms and practical problem solving skills using them.	BT	1																	
Analyze average and worst-case running times of given algorithm.	BT	1																	
Students should gain a good understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems	BT	2																	
Analyze and design algorithms on further advanced topics such as computational geometry, operations research, cryptography, number theoretic, algorithms etc.	BT	1																	
* 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 course
	2	Logic and Proving Techniques
Week 2	3	Asymptotic order of growth
	4	Recursive Exponentiation
Week 3	5	Sorting Algorithm: Brute Force Approach
	6	Design of Algorithms using Brute Force Approach
Week 4	7	Designing Algorithms using Divide-and-Conquer Multiplication, recurrence relations
	8	Generating Permutations, 0-1 Knapsack Algorithm Analysis
Week 5	9	Greedy Algorithms: Analysis of Merge-sort and Quick Sort Algorithm
	10	Substitution Method, Recursion-Tree Method, Master Therom
Week 6	11	Dynamic Programming
	12	Dynamic Programming for Solving Optimization Problems
Week 7	13	Linear Vs Non-linear Data Structures
	14	Matrix Chain Multiplication Problem
Week 8	1 hours	Mid Term
Week 9	15	Graph Theory and Types of Graphs
	16	BFS and DFS
Week 10	17	Paths in Graphs
	18	Breadth-first search, Dijkstra's algorithm,
Week 11	19	heaps, Bellman-Ford algorithm
	20	shortest paths in DAGs
Week 12	21	Greedy Algorithms
	22	Minimum spanning trees, Huffman encoding,
Week 13	23	Horn formulas,
	24	set cover (approximation algorithm)
Week 14	25	Dynamic Programming
	26	Shortest paths in DAGs (revisited),
Week 15	27	longest increasing subsequence
	28	edit distance, knapsack problem
Week 16	29	Complexity Theory
	30	Turing machine
Week 17	2 hours	Final Term

Laboratory Projects/Experiments Done in the Course

Programming Assignments Done in the Course

Instructor Name Dr. Tahir Alyas

Instructor Signature

Date