COURSE DESCRIPTION FORM

INSTITUTION National University of Computer and Emerging Sciences

PROGRAM (S) TO BE EVALUATED Computer Science

A. Course Description

(Fill out the following table for each course in your computer science curriculum. A filled out form should not be more than 2-3 pages.)

Course Code	CS302
Course Title	DESIGN & ANALYSIS OF ALGORITHMS
Credit Hours	3
Prerequisites by Course(s) and Topics	Data Structures
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Assessment with the weight.

Assessment Type	**Weight**
Assignments / Quizzes	15 (10+5)
Mid-Terms	25 (12.5 each)
Project	10
Final	50

Course Coordinator	Dr. Muhammad Atif Tahir
URL (if any)	
Current Catalog Description	Introduction; role of algorithms in computing, Analysis on nature of input and size of input Asymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω, Sorting Algorithm analysis, loop invariants, Recursion and recurrence relations; Algorithm Design Techniques, Brute Force Approach, Divide-and-conquer approach; Merge, Quick Sort, Greedy approach; Dynamic programming; Elements of Dynamic Programming, Search trees; Heaps; Hashing; Graph algorithms, shortest paths, sparse graphs, String matching; Introduction to complexity classes;
Textbook (or **Laboratory Manual** for Laboratory Courses)	Thomas H. Cormen et al. “Introduction to Algorithms” 2nd Edition. Anany Levitin “Introduction to the design and analysis of algorithms” 3rd Edition.
Reference Material	1. Dasgupta, Papadimitriou, and Vazirani: *Algorithms*, McGraw-Hill, 2006

2. Kleinberg, Tardos: *Algorithm Design*, Addison Wesley Longman, 2006.
3. Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos,
4. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne
5. ACM Transactions on Algorithms

Course Goals

A. Course Learning Outcomes (CLOs)

1. Design algorithms using different algorithms design techniques i.e. Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms
2. Analyse the time and space complexity of different algorithms by using standard analysis techniques for recursive and non-recursive algorithms.
3. Discussion on Asymptotic notations, standard complexity classes and representation of time complexities in asymptotic notations of standard complexity functions
4. Describe, compare, analyse, and solve general algorithmic problem types: Sorting, Searching, String Processing, Graph.
5. Implement the algorithms, compare the implementations empirically, and apply fundamental algorithms knowledge to solve real-world problems.
6. Understanding of NP-Completeness and Approximate Problems.

B. Program Learning Outcomes

For each attribute below, indicate whether this attribute is covered in this course or not. Leave the cell blank if the enablement is little or non-existent.

1. Academic Education:	To prepare graduates as computing professionals	✔
2. Knowledge for Solving Computing Problems:	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.	✔
3. Problem Analysis:	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.	✔
4. Design/ Development of Solutions:	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	✔

5. Modern Tool Usage:	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.	
6. Individual and Team Work:	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.	✔
7. Communication:	Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.	
8. Computing Professionalism and Society:	Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.	
9. Ethics:	Understand and commit to professional ethics, responsibilities, and norms of professional computing practice.	
10. Life-long Learning:	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.	

C. Relation between CLOs and PLOs
(CLO: Course Learning Outcome, PLOs: Program Learning Outcomes)

		PLOs									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CLOs	1	✔	✔		✔		✔				
	2	✔	✔	✔			✔				
	3	✔	✔	✔			✔				
	4	✔	✔		✔		✔				
	5	✔	✔		✔		✔				
	6	✔	✔	✔			✔				

Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)

1. Topics to be covered:

List of Topics	No. of Weeks	Contact Hours	CLO
Basics of Algorithms, Mathematical Foundation, Growth of Function, Asymptotic Notations.	**2**	**6**	**2,3**
Divide and Conquer, Substitution Method, Recurrence-Tree Method, Master's Method.	**2**	**6**	**2,3**
Sorting (Merge, Insertion, Quick, Heap, Counting, Radix), Data Structures (Stack, Queue, Linked List, Hash Table, Binary Tree).	**1**	**3**	**1,2,3,4**
Dynamic Programming	**1.5**	**4.5**	**1,2**
Greedy Algorithms, Graph Theory (Graph Categorization, Graph Terminology, Representation of Graphs, BFS & DFS, Strongly Connected Components, Greedy Algorithms: Kruskal's Algorithm, Prim's Algorithms, Bellman-Ford Algorithms, Dijkstra's Algorithm)	**3**	**9**	**2, 4**
Geometric Algorithms (Introduction, Graham Scan, Close Points). String Matching	**2**	**6**	**2,4,5**
NP Complete Problems and Solutions using Approximation Algorithm	**2**	**6**	**2,6**
Review	**0.5**	**1.5**	**1,4,6**
Project Presentations	**1**	**3**	**1,2,3,4,5**
Total	**15**	**45**	

Laboratory Projects/Experiments Done in the Course

Programming Assignments Done in the Course

Yes, Algorithm published in research paper based on Genetic Algorithm was implemented using a preferred programming language selected by student (e.g. C++, Java, Python)

Class Time Spent on (in credit hours)

Theory	**Problem Analysis**	**Solution Design**	**Social and Ethical Issues**
30	10	5	0

Oral and Written Communications

Every student is required to submit at least __1___ written reports of typically _2____ pages and to make __1___ oral presentations of typically __10___ minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Instructor Name: Dr Muhammad Atif Tahir / Mr Muhammad Waqas / Mr Zeshan Khan

Instructor Signature

Dated: Fall 2019