COURSE DESCRIPTION FORM: CL-2001 Data Structures - Lab

INSTITUTION FAST School of Computing, National University of Computer and Emerging Sciences, Islamabad Campus

BS-CS: **Fall - 2021**

PROGRAM TO BE EVALUATED

Course Description

Lab Course Code	CL-2001
Lab Course Title	Data Structure - Lab
Credit Hours	1
Lab Course Instructors	Mr. Muhammad Toqeer, Ms. Madiha, Ms. Urooj Ghani, Ms. Riva Malik, Ms. Ayesha Kamran
Grading Policy	Absolute Grading (Same grade as theory)
Policy about missed assessment items in the course	Retake of missed assessment items (other than sessional/ final exam) will not be held. Student who misses an assessment item (other than sessional / final exam) is awarded zero marks in that assessment item i.e. late submission will not be accepted. For missed sessional/ final exam, exam retake/ pre-take application along with necessary evidence are required to be submitted to the department secretary. The examination assessment and retake committee decides the exam retake/ pre-take cases.
Course Plagiarism Policy	**Plagiarism in a project or final exam will result in F grade in the course.** **Plagiarism in a lab task will result in zero marks in the whole lab tasks category.**
Prerequisites by Course(s) or Topics	Object Oriented Programming
Assessment Instruments with Weights (Sessional exam, final exam, lab tasks, lab project, etc.)	Assessment with the weight. **Assessment Type** — **Weight** Lab Tasks (>12) — 15 weight (in theory)
Lab Course Coordinator	Muhammad Toqeer (Lab)
URL (if any)	
Course Catalog Description	Arrays, Link lists, Stacks, Queue, Binary Trees, Graphs, Hashing, Heap
Laboratory Manual	Uploaded on LMS

Course Learning Outcomes

A. Course Learning Outcomes (CLOs)

During this course, students learn strategies and techniques to efficiently store data (Data Structures) and to perform processing on such data in efficient ways (Algorithms), as well as on the analysis and design of such techniques. After successful completion of this course students should be able to:

1. Define basic static and dynamic data structures and relevant standard algorithms for them: Stacks, Queues, Linked lists, Trees, Graphs, Heap, Priority queue, and sorting/searching algorithms
2. Demonstrate advantages and disadvantages of specific algorithms and data structures.
3. Prepare the students to pick the right data structure for a given problem and understand which data structure to implement in a certain scenario.
4. Determine and demonstrate bugs in program, recognize needed basic operations with data structures.
5. Formulate new solutions for programing problems or improve existing code using learned algorithms and data structures.
6. Evaluate algorithms and data structures in terms of time and memory complexity of basic operations.
7. Apply concepts learned in various domains like DBMS, compiler construction etc.

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. Computing Knowledge:	Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems.	✔
kkvkvks		✔

2. Problem Analysis:	Identify, formulate, research literature, and analyze complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences.			
3. Design/ Develop Solutions:	Design solutions for complex computing problems and design systems, components, and processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	✔		
4. Investigation & Experimentation	Conduct investigation of complex computing problems using research based knowledge and research based methods.			
5. Modern Tool Usage:	Create, select, and apply appropriate techniques, resources and modern computing tools, including prediction and modelling for complex computing problems.	✔		
6. Society Responsibility:	Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues relevant to context of complex computing problems.			
7. Environment and Sustainability:	Understand and evaluate sustainability and impact of professional computing work in the solution of complex computing problems.			
8. Ethics:	Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.			
9. Individual and Team Work:	Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.			
10. Communication:	Communicate effectively on complex computing activities with the computing community and with society at large.			

11. Project Management and Finance:	Demonstrate knowledge and understanding of management principles and economic decision making and apply these to one's own work as a member or a team.
12. Life Long Learning:	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological changes.

C. Mapping of CLOs on PLOs
(CLO: Course Learning Outcome, PLOs: Program Learning Outcomes)

		PLOs											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CLOs	1	✔	✔	✔									
	2	✔	✔	✔									
	3	✔	✔	✔									
	4	✔	✔	✔									
	5	✔	✔	✔									
	6	✔	✔	✔		✔							
	7	✔				✔							

Weeks	Contents/Topics	Assessment Items (Case Study/ Exercise Assignment/ Quiz etc.)	CLO (s)
Week-01	ADT, Templates	Lab task 1	1,2,3
Week-02	Templates with Unit testing	Lab task 2	1,2,3
Week-03	2D/3D Arrays with Unit testing	Lab task 3	1,2,3
Week-04	List (Array-based), Recursion with Unit testing	Lab task 4	1,2,3
Week-05	Linked lists (Singly), Recursion with Unit testing	Lab task 5	1,2,3
Week-06	Linked list variations (Circular, double), Recursion with Unit testing	Lab task 6	1,2,3,5
Week-07	Queue with Unit testing	Lab task 7	1,2,3
Week-08	Stacks & Applications with Unit testing	Lab task 8	1,2,3
Week-9	Binary Search Trees (BST) with Unit testing	Lab task 9	1,2,3
Week-10	Binary Search Trees (BST) with Unit testing	Lab task 10	1,2,3,5
Week-11	**Second Sessional Exam**		
Week-12	AVL Trees with Unit testing	Lab task 11	1,2,3,5
Week- 13	Graph ADT, Traversals with Unit testing	Lab task 12	1,2,3
Week- 14	Hashing with Unit testing	Lab task 13	1,2,3
Week- 15-16	Project		

Practical/ Programming Work/ Tools	Visual Studio 2019 with C++ and GTEST			
Lab Time Spent (in percentage)	**Theory**	**Problem Analysis & Design**	**Implementation**	**Social and Ethical Issues**
	5	15	75	5