COURSE DESCRIPTION FORM

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

PROGRAM(S) TO BE EVALUATED **BS-CS FALL 2022**

Course Description

Course Code	CS-1002
Course Title	Programming Fundamentals
Credit Hours	3+1
Prerequisites by Course(s) and Topics	N/A
Grading Policy	Absolute grading
Policy about missed assessment items in the course	Retake of missed assessment items (other than midterm/ final exam) will not be held. For a missed midterm/ final exam, an exam retake/ pretake application along with necessary evidence are required to be submitted to the department secretary. The examination assessment and retake committee decides the exam retake/ pretake cases.
Policy about late submission of assessment item	Penalties for late submissions of assignment/project: 1. Up to 3 hours late, loss of 40% of the mark awarded. 2. After 3 hours, assignment/project will not be accepted for marking.
Course Plagiarism Policy	Plagiarism in project or midterm or final exam may result in F grade in the course. Plagiarism in an assignment/project will result in zero marks in the whole assignment category for both students.
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Assessment items of Theory Part

Assessment Item	Number	Weight (%)
Assignments	4	12
Sessional - I	1	10
Sessional - II	1	15
Quizzes	>= 4	10
Project	1	13
Final Exam	1	40

Assessment items of Lab Part

Assessment Item	Number	Weight (%)
Lab Tasks	>=14	30
Project	1	10
Sessional - I	1	8

	<table><tr><td>Sessional - II</td><td>1</td><td>12</td></tr><tr><td>Final Exam</td><td>1</td><td>40</td></tr></table>
Course Instructors	Dr. Muhammad Aleem, Dr. Akhtar Jamil, Mr. Shehreyar Rashid, Ms. Ifrah Qaisar
Lab Instructors (if any)	Muhammad Toqeer, Riva Malik, M. Usman Khan, Sher Bano
Course Coordinator	Dr. Muhammad Aleem, Dr. Akhtar Jamil
URL (if any)	https://classroom.google.com/c/NTI2NTQ5MTM4Nzk5?cjc=kts6kea Class code: kts6kea
Current Catalog Description	The course aims to equip students with the basic computing concepts and to provide them the ability to analyze the given requirements for solving problems in different domains while implementing the solutions on a computer system. It emphasizes on developing an algorithm and applying the basic programming constructs like control structures, arrays, functions, pointers, dynamic memory allocation, etc. for its development. The students will learn the syntax of the C++ programming language for the implementation.
Textbook (or **Laboratory Manual** for Laboratory Courses)	Tony Gaddis "STARTING OUT WITH C++" 9th Edition
Reference Material	Paul Deitel, Harvey Deitel "C++ How to Program" 10th Edition Walter Savitch "Problem Solving with C++" 10th Edition

Course Learning Outcomes

A. Course Learning Outcomes (CLOs)

After completion of the course, the students shall be able to:

1. **Understand** basic problem-solving steps and logic constructs.
2. **Apply** basic programming concepts.
3. Design and implement algorithms to **solve** real-world 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. Computing Knowledge:	Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems	✔
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 Teamwork:	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 of a team.	
12. Lifelong 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			✔						✔			

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

List of Topics	No. of Weeks	Contact Hours
- Problem-solving, Basic flowchart, block diagram, and programming languages. - Primitive data types, input/output (hello world). - Signed and unsigned data types, constants and variables.	**1**	**3**
- Arithmetic operators (+, -, *, /, % and their compound counterparts) with their associativity and precedence. - Bit wise operators	**2**	**6**
- Conditional/selection structures. - Comparison and logical operators. - if, if. . .else and if else if structure. - Switch statement, *break* statement. - Ternary operator.	**2**	**6**
1st Sessional Examination		
- Repetition structures. - Pre/post increment/decrement operators. - while loop (sentinels + condition). - Loop with *for.* - Loop with *do-while.* - Nesting of *while*, *for* loop and *continue* statement.	**3**	**9**
- Function prototypes, definition, and calling.	**1**	**3**
- Introduction to Arrays. - Array initialization and representation. - Char arrays. - Multi-Dimensional Arrays (MDA). - MDA representation in memory.	**1.33**	**4**
2nd Sessional Examination		
- Aliases, parameters passing by value and by reference (passing arrays). - Function calling order and stack (function within a function). - Recursion	**1.66**	**5**
- Pointers. - const. vs. non-const. pointers, a pointer to const. data vs. a pointer to non-constant data. - Using pointers. - Dynamic memory allocation. - Array of pointers.	**2**	**6**
- Header files (creating own file). - Files handling - Opening flags (app mode).	**1**	**3**

	Total	**15**	**45**	
Laboratory Projects/Experiments Done in the Course	Yes, there are lab tasks with every lab of three hours.			
Programming Assignments Done in the Course	Yes, there are six programming assignments and a project.			
Class Time Spent (in hours)	**Theory**	**Problem Analysis**	**Solution Design**	**Social and Ethical Issues**
	34	5	5	1
Oral and Written Communications	Every student is required to submit at least __1___ written reports of typically ___5__ pages and to make __1___ demonstration of typically ____10___ minutes duration.			

Lab/ Practical Component of the course

Weeks	List of Topics	Assessment Items (Case Study/ Exercise Assignment/ Quiz etc.)
Week-01	Ubuntu installation, Shell commands	
Week-02	Pseudo code, Scratch tool	
Week-03	Variable and data Types in C++ Operators (arithmetic)	
Week-04	Operators (arithmetic, bitwise)	
Week-05	Conditional structures-I (if-else, ternary operators)	
Week-06	Sessional - I	
Week-07	Conditional structures-II (Switch case)	
Week-08	Repetitions-I (while loop, for loop)	
Week-09	Repetitions-II (do-while loop, nested while loop, nested for loop)	
Week-10	Functions (definition, calling, forward declaration)	
Week-11	Arrays-I (1D arrays)	
Week-12	Arrays-II (char and multi-dimensional arrays)	
Week 12	Sessional – II	
Week-13	Functions (parameter passing by value/reference)	
Week-14	Introduction to pointers and dynamic memory allocation (for 1D)	
Week-15	Dynamic memory allocation (2D, and 3D)	
Week-16	Basic File Handling	

Practical/ Programming Work/ Tools: Ubuntu, g++