COURSE DESCRIPTION FORM

INSTITUTION National University of Computers and Emerging Sciences

BS Computer Science

PROGRAM (S) TO BE EVALUATED

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	CL1002
Course Title	Programming Fundamentals Lab
Credit Hours	1
Prerequisites by Course(s) and Topics	None
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	**Lab Tasks:20 (Best 10: 2 each)** **Mid: 20** **Project: 10** **Final: 50**
Course Coordinator	Basit Ali
URL (if any)	
Current Catalog Description	
Textbook (or **Laboratory Manual** for Laboratory Courses)	Name: C How to Program with an Introduction to C Global Edition - 7th Edition Authors: Paul Deitel, Harvey Deitel Publisher: Pearson Name: Problem Solving and Program Design in C - 7th Edition Authors: Maureen Sprankle , Jim Hubbard Publisher: Prentice Hall
Reference Material	Name: Working with C / Let us C

	Author(s): YashwantKanetkar Publisher: BPB Publications Name: Waite Group's Turbo C - Programming for the PC Authors: Robert Lafore Publisher: SAMS
Course Goals	

A. Course Learning Outcomes (CLOs)	Level
CLO 1: Describe fundamental concepts of structured and procedural programming, use pseudo-codes and simple programs to understand control structures, iterative structures and functions using C language.	**C3, PLO1**
CLO 2: Examine code writing, compiling, debugging and program execution.	**C3, PLO5**
CLO 3: Justify problem solving techniques and analytical thinking by identifying the concepts and properties of algorithms.	**C5, PLO2**
CLO 4: Design basic problems of the real world through small/medium size programs given as course projects.	**C6, PLO5**

B. Program learning outcomes (PLO)			
PLO 1	Computing Knowledge	Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems.	☐
PLO 2	Problem Analysis	Identify, formulate, research literature, and analyse complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences.	☐
PLO 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.	
PLO 4	Investigation& Experimentation	Conduct investigation of complex computing problems using research-based knowledge and research-based methods	
PLO 5	Modern Tool Usage	Create, select, and apply appropriate techniques, resources and modern computing tools, including	☐

		prediction and modelling for complex computing problems.	
PLO 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.	
PLO 7	Environment and Sustainability	Understand and evaluate sustainability and impact of professional computing work in the solution of complex computing problems	
PLO 8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.	
PLO 9	Individual and Teamwork	Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.	
PLO 10	Communication	Communicate effectively on complex computing activities with the computing community and with society at large.	
PLO 11	Project Mgmt. 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.	
PLO 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. Relation between CLOs and 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					☒							

Topics Covered in the Course, with Number of Lectures on Each

Week	**Topics**	**CLO**	**Assessment**

Topic (assume 16-week instruction and three-hour lectures)	**Week 1**	Introduction to Programming Fundamentals, Introduction to problem solving (Real-World Examples), PAC charts, IPO Charts, FlowCharts, Pseudocode writing. (Attempt charts on either paper or utility (diagrams.net) for creation)	**Lab 01**		
	Week 2	1. Daily life real world problems 2. Flowchart in depth & Extended flowchart & Algorithms 3. Working examples 4. Introduction to GitHub(Login, signing)	**Lab 02**	***Announce Project (Week 3)***	
	Week 3	1. Problem solving with decisions, Basic Computer Organization, Intro to IDE (compiled program, text editors, debuggers, etc.), Library, Linking, Compiling & Loading. Program structure and Execution, First Program with Input and Output, Constant, Variables, Keywords, Escape sequence, Format Specifiers, Data types, Data manipulation. 2. More on GitHub, Markdown language, commits, branching, pull and merge requests	**Lab 03**		
	Week 4	Decision Structure (if, if- else, else if and Switch Statements)	**Lab 04**		
	Week 5	Nested if else, nested structures, Operators (Logical, Conditional, Bitwise, Modulus)	**Lab 05**		
	Week 6	**THEORY MID I Examination**			
	Week 7	1. Basic loops: for, while and do-while 2. Referencing 3. Introduction to pointer (Just referencing)	**Lab 06**		
	Week 8	Loops with 1D arrays	**Lab 07**		
	Week 9	**Lab Mid**	**Lab Mid**		
	Week 10	Nested Loops with N-D arrays	**Lab 08**		
	Week 11	**THEORY MID II Examination**			
	Week 12	1. Functions: Declaration, Definition and Calling, passing values to functions, Passing arrays to functions 2. Standard library string functions 3. 2D array of characters	**Lab 09**		

Week 13	1. Recursion (types, stack calling) 2. Constant & Static 3. Introduction to Structures	**Lab 10**		
Week 14	1. Nested Structure, Composition and Structure array 2. Filing in C	**Lab 11**		
Week 15	1. Single Pointer(including structure) with DMA 2. 2D pointers with DMA	**Lab 12**		
Week 16	**Project Submission**			

Laboratory Projects/Experiments Done in the Course	There will be weekly labs starting from the first week. The following is a summary of the Lab exercises given to Students. ● Introduction to Problem solving statements. ● Introduction To Conditional Statement In C ● Control Structure (Repetition) ● GitHub ● Functions and Recursion. ● Arrays (1D, 2D, 3D) ● String sorting and searching algorithms. ● Pointers ● Dynamic memory allocation ● Structures ● Filing in C
Programming Assignments Done in the Course	Assignment related to Functions, Arrays, Pointers, Structures, Dynamic Memory and File Processing will be done

Class Time Spent on (in credit hours)	**Theory**	**Problem Analysis**	**Solution Design**	**Social**
	15%	50%	30%	5%

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 _____

Instructor Signature _____

Date _____