



## 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.)

<b>Course Code</b>	CS1002
<b>Course Title</b>	Programming Fundamentals
<b>Credit Hours</b>	3+1
<b>Prerequisites by Course(s) and Topics</b>	None
<b>Assessment Instruments with Weights</b> (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	<b>Mid-1:15</b> <b>Mid-2:15</b> <b>Assignment: 10 (Three Assignments)</b> <b>Quizzes: 10 (Three Quizzes)</b> <b>Final:50</b>
<b>Course Coordinator</b>	Muhammad Shahzad
<b>URL (if any)</b>	
<b>Current Catalog Description</b>	
<b>Textbook (or Laboratory Manual for Laboratory Courses)</b>	<u>Name:</u> C How to Program with an Introduction to C Global Edition - 7th Edition <u>Authors:</u> Paul Deitel, Harvey Deitel <u>Publisher:</u> Pearson  <u>Name:</u> Problem Solving and Program Design in C - 7th Edition <u>Authors:</u> Maureen Sprankle , Jim Hubbard <u>Publisher:</u> Prentice Hall
<b>Reference Material</b>	<u>Name:</u> Working with C / Let us C

	<u>Author(s):</u> YashwantKanetkar <u>Publisher:</u> BPB Publications  <u>Name:</u> Waite Group's Turbo C - Programming for the PC <u>Authors:</u> Robert Lafore <u>Publisher:</u> SAMS		
Course Goals	A. Course Learning Outcomes (CLOs)		
	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	



	<div></div> <div>12</div> <div>11</div> <div>10</div> <div>9</div> <div>8</div> <div>7</div> <div>6</div> <div>5</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> <div>CLOs</div> <div>4</div>				
Topics Covered in the Course, with Number of	<table><tr><th>Week</th><th>Topics</th><th>CLO</th><th>Assessment</th></tr></table>	Week	Topics	CLO	Assessment
Week	Topics	CLO	Assessment		

<b>Lectures on Each Topic</b> (assume 15-week instruction and one-hour lectures)	<b>Week 1</b>	Discussion of the course outline, Introduction to problem solving, what is algorithm, how to write pseudo code, programming structures, problem solving with the sequential structures and, Basic Flowchart, IPO and PAC	<b>1</b>	
	<b>Week 2</b>	Problem solving with decisions and iterative structures, Basic Computer Organization, Intro to IDE (compiled program, text editors, debuggers, etc.), Program structure and Execution, First Program with Input and Output	<b>1,2</b>	<b>Project Announcement</b>
	<b>Week 3</b>	1. Constant, Variables, Keywords, Escape sequence 2. Format Specifiers, Data types, Data manipulation 3. Library, Linking, Compiling & Loading	<b>2,3</b>	<b>Quiz no 1 Assignment 1 Friday Release Week 3</b>
	<b>Week 4</b>	1. Decision Control Structures: If statements and if-else statement 2. Basic switch statements 3. Some working examples	<b>2,3</b>	
	<b>Week 5</b>	1. Nested if statements & switch statements 2. Logical & Conditional Operators 3. Working examples	<b>2,3,5</b>	<b>Assignment 1 submission Monday Week 5</b>
	<b>Week 6</b>	<b>MID I Examination</b>		<b>Quiz no 2</b>
	<b>Week 7</b>	1. Introduction to Loops Design 2. For, while and do-while loops 3. Some working examples	<b>2,3,5</b>	<b>Assignment 2 Friday Release Week 7</b>
	<b>Week 8</b>	1. Nested Loops 2. Break and Continue Statement 3. Working examples	<b>2,3,4,5</b>	
	<b>Week 9</b>	1. Introduction to 1D Arrays 2. Multiple subscripted arrays 3. Working examples	<b>2,3,4,5</b>	<b>Assignment 2 submission Monday Week 9</b>
				<b>Quiz no 3</b>

	<b>Week 10</b>	1. Functions: Declaration, Definition and Calling, passing values to functions, Passing arrays to functions 2. Standard library string functions 3. 2D array of characters	<b>2,3,4,5</b>	
	<b>Week 11</b>	<b>MID II Examination</b>		
	<b>Week 12</b>	1. Recursion 2. Introduction to Structures and Structure array 3. Working examples	<b>3,4,5</b>	<b>Assignment 3 Friday Release Week 12</b>
	<b>Week 13</b>	1. Nested structures, Passing structure function 2. Filing in C 3. Introduction to pointers	<b>3,4,5,6</b>	
	<b>Week 14</b>	1. Pointers and Arrays 2. Dynamic memory allocation 3. Void pointers 4. Examples	<b>2,3,4,5 ,6</b>	<b>Assignment 3 submission Monday Week 14</b>
	<b>Week 15</b>		<b>3,5</b>	<b>Project Submissions in 12<sup>th</sup> LAB and Finalization of Sessional marks</b>
		<b>Revision</b>		
	<b>Week 16</b>	<b>Final Exam</b>		
<b>Laboratory Projects/Experiments Done in the Course</b>	<b>Week 1</b>	Problem solving with sequential structure using Scratch		
	<b>Week 2</b>	Problem solving with the decision and iterative structures using Scratch		
	<b>Week 3</b>	Introduction to IDE and Basic Programming Constructs		<b>Announce Project</b>
	<b>Week 4</b>	Introduction of operators and math.h library functions		
	<b>Week 5</b>	Basic Decision Structure (if, if- else and Switch Statements)		

	<b>Week 6</b>	<b>THEORY MID I Examination</b>			
	<b>Week 7</b>	Nested Decision Structures			
	<b>Week 8</b>	Iterative Statements in C			
	<b>Week 9</b>	<b>Lab Mid</b>			
	<b>Week 10</b>	Nested Iterations, Arrays Multiple Dimension Array (2D,3D) in C			
	<b>Week 11</b>	<b>THEORY MID II Examination</b>			
	<b>Week 12</b>	Functions, Strings and Recursion			
	<b>Week 13</b>	Introduction to Structures & Nested Structure			
	<b>Week 14</b>	Introduction to file processing and basic operations on files and Introduction to Pointers			
	<b>Week 15</b>	Accessing Arrays using pointer Dynamic Memory Management			
	<b>Week 16</b>	<b>Project Submission</b>			
	<b>Programming Assignments Done in the Course</b>	Assignment related to Functions, Arrays, Pointers, Structures, Dynamic Memory and File Processing will be done			
<b>Class Time Spent on</b> (in credit hours)	<b>Theory</b>	<b>Problem Analysis</b>	<b>Solution Design</b>	<b>Social</b>	
	15%	50%	30%	5%	
<b>Oral and Written Communications</b>	Every student is required to submit at least <u>1</u> written reports of typically <u>2</u> pages and to make <u>1</u> oral presentations of typically <u>10</u> 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** \_\_\_\_\_