National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Electrical Engineering

Fundamentals of Computer Programming

Course Code:	CS110	**Semester:**	1st
Credit Hours:	3+1	**Prerequisite Codes:**	None
Instructor:	Dr. Anis ur Rahman	**Class:**	BSCS
Office:	A216, Faculty block	**Telephone:**	NA
Lecture Days:	Mons, Tues & Fris	**E-mail:**	anis.rahman@seecs.edu.pk
Class Room:	TBA	**Consulting Hours:**	9:00 to 10:00
Lab Engineer:	Rabbia Hassan	**Lab Engineer Email:**	rabbia.hassan@seecs.edu.pk
Knowledge Group:	Programming	**Updates on LMS:**	After every lecture

Course Description:

The course introduces the fundamental concepts underlying modern computer programming. A systematic approach is used to teach students how to write programs that solve well specified problems. Emphasis is placed on the mastery of basic programming skills, with a considerable attention to the fundamental building blocks of computer programs, and the associated concepts and principles. The essentials of sequential processing and control flow are taught in a procedural programming context prior to introducing classes, objects and related object-oriented programming concepts. To ensure the development of the necessary competencies, assigned homework includes the development of program solutions to problems of adequate complexity and relevance.

Course Objectives:

The learning objectives are:

1. Developing comprehensive knowledge about the fundamental principles, concepts and constructs of modern computer programming.
2. Developing competencies for the design, coding and debugging of computer programs.

Course Learning Outcomes (CLOs):		
Upon completion of the course, it is expected that you will be able to:	**PLO**	**BT Level***
1. Describe the fundamental programming constructs and articulate how they are used to develop a program with a desired runtime execution flow.	1	C2
2. Develop programs to implement computer-based solutions of well specified problems.	2	C1, C2
3. Distinguish the advantages and limitations resulting from the use of different language constructs that embody similar programming concepts.	3	C6
4. Articulate where computer programs fit in the provision of computer-based solutions to real world problems.	5	C3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Mapping of Course Learning Outcomes (CLOs) to Program Learning Outcomes (PLOs)

PLOs/CLOs	CLO1	CLO2	CLO3	CLO4	Emphasis Level
PLO a (Computing Knowledge)	×				1
PLO b (Problem and Requirement Analysis)		×			3
PLO c (Design, Implementation and Evaluation of Solutions)			×		2
PLO d (Individual and Team Work)					
PLO e (Professional and ethical Responsibility)					
PLO f (Communication)					
PLO g (Local and Global Computing Impact Analysis)					
PLO h (Lifelong Learning)					
PLO i (Modern tool usage)				×	2
PLO j (Design Choices and Tradeoffs Analysis)			×		2
PLO k (Adherence to Design and Development Principles)		×	×		1

Mapping of CLOs to Assessment Modules and Weightages (In accordance with NUST statutes)

Assessments/CLOs	CLO1	CLO2	CLO3	CLO4
Theory: 75%				
• Quizzes: 15%				
• Assignments: 10%				
• OHT-1: 15%				
• OHT-2: 15%				
• End Semester Exam: 45%				
Practical: 25%				
• Labs Assignments: 70%				
• Semester Project: 30%				
Total : 100 %				
To be filled in at the end of the course.				

Books:

Text Book:
1. Paul J. Deitel and Harvey M. Deitel, C: How to Program, Prentice Hall, 2010.

Reference Books:
- The C Programming Language (2nd Ed.) by Kernighan and Ritchie, 1988.
- Code Complete (2nd Ed.) by Steve McConnell, 2004.
- The Art of Computer Programming (TAOCP) by Donald E. Knuth, 1968.

Topics to be Covered:	
1. Problem Solving	2. Algorithms
3. Data Types and Calculations	4. Decision
5. Repetition	6. Structured Programming
7. Arrays & Files	8. Objects and Classes
9. Special Topics	

Lecture Breakdown:

Week No.	Topics	Assessment	Remarks
1	Lecture 1: introduction Lecture 2: history of programming languages Lecture 3: algorithms Lab 01		
2	Lecture 4: flowcharts and pseudocode Lecture 5: overview of C program Lecture 6: writing, compiling and debugging C programs Lab 02		
3	Lecture 7: coding style Lecture 8: statements Lecture 9: variables and datatypes Lab 03		
4	Lecture 10: operators and expressions Lecture 11: selection Lecture 12: relational operators Lab 04		
5	Lecture 13: repetition Lecture 14: switch, break, continue Lecture 15: conditional operators Lab 05		
6	**OHT-1**		
7	Lecture 16: logical operators Lecture 17: modular programming Lecture 18: scope of variables Lab 06		
8	Lecture 19: function definition and prototypes Lecture 20: call by value, reference Lecture 21: recursion Lab 07		
9	Lecture 22: default arguments Lecture 23: inline functions Lecture 24: command-line arguments Lab 08		
10	Lecture 25: pointers and memory addressing Lecture 26: arrays Lecture 27: strings Lab 09		
11	Lecture 28: arrays and pointer arithmetic Lecture 29: user-defined structures Lecture 30: struct, unions and bitfields Lab 10		
12	**OHT-2**		
13	Lecture 31: memory allocation		

	Lecture 32: pointers to pointers Lecture 33: pointer and string arrays Lab 11
14	Lecture 34: multidimensional arrays Lecture 35: void and function pointers Lecture 36: files and streams Lab 12
15	Lecture 37: file pointers Lecture 38: error handling Lecture 39: data structures Lab 13
16	Lecture 40: link lists Lecture 41: stacks and queues Lecture 42: classes/objects Lab 14
17	Lecture 43: fields, instance and methods Lecture 44: constructors and encapsulation Lecture 45: inheritance and polymorphism Lab 15
18	**ESE**

Lab Experiments:	
Lab 01:	Introduction to Programming and the Translation Process
Lab 02:	Introduction to the C Programming Language
Lab 03:	Expressions, Input, Output and Data Type Conversions
Lab04:	Conditional Statements
Lab 05:	Looping Statements
Lab 06:	Introduction to Void Functions
Lab 07:	Functions that Return a Value
Lab 08:	Arrays
Lab 09:	Searching and Sorting Arrays
Lab 10:	Pointers
Lab 11:	Characters and Strings
Lab 12:	Structures and Abstract Data Types
Lab 13:	Advanced File Operations
Lab 14:	Introduction to Classes
Lab 15:	Project Demos

Tools / Software Requirement:
MS Visual Studio gcc (Cygwin with Notepad++)

Grading Policy:	
Quiz Policy:	The quizzes will be unannounced and normally last for ten minutes. The question framed is to test the concepts involved in last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion.
Assignment Policy:	In order to develop comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams.
Lab Conduct:	The labs will be conducted for three hours every week. A lab handout will be given in advance for study and analysis The lab handouts will also be placed on LMS. The students are to submit their results by giving a lab report at the end of lab for evaluation. One lab report per group will be required. However, students will also be evaluated by oral viva during the lab.
Plagiarism:	SEECS maintains a zero tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SEECS plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action.