NATIONAL UNIVERSITY
of Computer & Emerging Sciences, Lahore

Department of Computer Science

CS1002 – Programming Fundamentals
Fall 2022

Instructor Name: Anosha Khan
Email address: anosha.khan@lhr.nu.edu.pk
Office Location/Number: Liberty Labs, M-024
Office Hours: Tue –Thu 10:00AM -1:00PM

TA Name: Muhammad Hamza
Email address: l190907@lhr.nu.edu.pk

Course Information

Program: BS (CS) / (SE) **Credit Hours:** 3 + 1 (Lab) **Course Type:** Core

Class Venue:

Course Description/Objectives/Goals:

- To introduce the notion of algorithms.
- To develop problem solving and logic building skills in students.
- To introduce the basic concepts of programming in C++, including basic data types, expressions, iterations, functions and arrays.

Course Learning Outcomes (CLOs):

At the end of the course students will be able to:	**Domain**	**BT* Level**
Understand basic problem solving steps and logic constructs	C	2
Apply basic programming concepts	C	3
Design and implement algorithms to solve real world problems and should be able to translate a problem statement into pseudo-code/C++ code	C	3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain **Bloom's taxonomy Levels:** 1. Knowledge, 2. Comprehension, 3. Application, 4. Analysis, 5. Synthesis, 6. Evaluation		

Course Textbook

1. C++ Programming: Program Design Including Data Structures, by D. S. Malik (8th Edition)
2. C++: How to Program? by Deitel & Deitel (9th Edition)

Additional references and books related to the course:

1. Starting out with C++ from control structures through objects by **Tony Gaddis** 8th Edition
2. Theory and Problems of Programming with C++ by John R. Hubbard, 2nd Edition
3. Programming and Problem Solving with C++, Nell Dale
4. www.learncpp.com

Tentative Weekly Schedule

Week 1 Problem Solving and Programming Introduction	**Lecture 1** Course introduction and logistics Introduction to Computers, Basic Computer architecture with an overview of memory as consisting of addressable storage locations for keeping data and program. A program as a sequence of instructions and the Fetch-Decode-Execute cycle Fundamental arithmetic and logical operations provided by a typical machine. Some interesting programs like spreadsheets, databases and intelligent game playing programs etc.	**Lecture 2** A brief introduction of programming languages and the idea of compiling, linking and loading. Introduction of some fundamental operations provided by a Basic/C++ like programming language with/without getting into exact C++ program structure details. These include • Idea of a variable with an understanding that a variable is a place in memory without discussing the internal representation of data. • Assignment of values to variable • Basic arithmetic and logical operations performed on variables. • Input and Output operations Writing some simple programs for performing calculations using the fundamental operations.
Week 2 Simple C++ Programs	**Lecture 3** The structure of a C++ program with a single main function and very brief explanation of #include and named spaces. A high level description of some built in C++ datatypes (int, float, double, char, bool), variables declaration, assignment operator, input, output. Writing the programs introduced in the first week using the C++ syntax. Compiling and running the program	**Lecture 4** A review of Lecture 2 and the Introduction of logical operations and the use of logical operation for conditional execution (IF statements). C++ Operators (Arithmetic, Logical, and Relational) Use of operators for different datatypes. Translating programs written using pseudocode or a flowchart into working C++code.
Week 3 Simple C++ Programs	**Lecture 5** Translating programs written using pseudocode or a flowchart into working C++code continued. A basic introduction of operator precedence and writing complex expressions as a sequence of simple intermediate expressions.	**Lecture 6** Programming exercises. Writing clean code using indentation and comments.
Week 4 Simple C++ Programs If/Else	**Lecture 7** Using a Nested selection structure. Programs with if/else statements	**Lecture 8** Programs with nested if/else statements
Week 5	**Lecture 9**	**Lecture 10**

Repetition Structures(Loops)	Repetitions using while, for and do while	Problem solving using repetition structures
MID-I		
Week 6 Nested Control Structures	**Lecture 11** Problem solving nested repetition structures	**Lecture 12** Problem solving using nested repetition structures
Week 7 **Functions**	**Lecture 13** Function definition and calling: parameters and return types; Global and local variables scope and life time.	**Lecture 14** Function Parameters: Pass by value and pass by reference. Stack rolling and unrolling**.**
Week 8 **Functions**	**Lecture 15** Top-Down Design of a program and its implementation using functions Built-in functions	**Lecture 16** Function Overloading. Functions with Default Parameters.
Week 9 **File Handling**	**Lecture 17** I/O from simple text Files	
Week 9 **Arrays Introduction and Repetition structure**	**Lecture 18** Define and use fixed sized arrays**.** Array organization in memory and element access using Array name and index. Initialization using member initializer list, and by using loops.	
Week 10 **Arrays Processing**	**Lecture 19** Printing data, taking input, Processing by index and by elements Find Min, Max, Avg, Equilibrium Index Reverse: All Elements, odd/eve elements and indices Search: Linear and Binary	**Lecture 20** Passing arrays to functions. Design different functions for input, output, search, reverse, Shifting and Rotation of elements: right and left Insert and delete elements from ordered list using shifting.
MID-II		
Week 11 **Arrays Processing**	**Lecture 21** Passing arrays to functions use of const. Sorting: Bubble Sort, Selection Sort	**Lecture 22** Sorting: Insertion Sort, Even/odd Sort Merging sorted arrays. Application: Sets, Union, Intersection, difference.
Week 12 **CStrings and character Arrays Processing**	**Lecture 23** I/O from simple text Files in arrays. Difference between Null terminated CStrings and character arrays. Storage of CStrings in character arrays and aggregate I/O.	**Lecture 24** Functions design: Find String length, Compare strings, Find substring and replace, Calculate frequency of specific characters Remove specific characters.
Week 13 **2D Arrays Processing**	**Lecture 25** Using built in CSrting functions. Use of built-in rand () function. 2-Dimensional Array and how it is organized in memory in row/col major order. Initialization using member initializer list, and by using loops. I/O and processing of elements in row/col major order.	**Lecture 26** Application: Store and process Students Quiz marks. Find Min, Max, Avg, column and row wise. Sorting: row wise or column wise, complete array by specific column or row.
Week 14 **2D Arrays**	**Lecture 27** Passing 2D arrays to functions: Complete,	**Lecture 28** Application: Matrices storage and

Processing	individual rows, or elements. Processing diagonals: reverse elements, print data of whole array.	processing Addition, Subtraction, Multiplication, Transpose, Check for Upper and lower triangular. Use of graphic libraries functions. Designing header files for user defined functions.
Week 15 **2D CStrings Processing**	**Lecture 29** Storage and processing of CStrings in 2D Arrays. Bitwise operators Binary files I/O **Application: Data compression.**	
Week 16 **Structures**	**Lecture 30** Introduction and using structs Passing and returning from functions	**Lecture 31** Arrays of Structs, Application of structs
Final Exam		

(Tentative) Grading Criteria:

1. Assignments + Homework **(10 %)**
2. Quizzes **(10 %)**
3. Midterms **(30 %)**
4. Project **(10 %)**
5. Final Exam **(40 %)**

- Grading scheme for this course is **Absolute** under application of CS department's grading policies.
- Minimum requirement to pass this course is to obtain at least **50%** absolute marks

Course Policies:

- All assignments and homework must be done individually.
- Late Submissions of assignments will not be accepted.
- **Plagiarism** in any work (Quiz, Assignment, Midterms, Project and Final Exam) from any source, Internet or a Student will result in **deduction of absolute marks or F** grade.
- Minimum **80%** attendance is required for appearing in the Final exams.