

Department of Computer Science

CS1002 – Programming Fundamentals FALL 2021

Instructor Name: Aamina Batool TA Name: Salman Muzammil

Email address: Amina.batool@nu.edu.pk Email address: limailto:line.nu.edu.pk

Office Location/Number: M-107
Office Hours: TBA

Course Information

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

Class Meeting Time: 1:00 pm – 2:20 pm (Monday & Wednesday)

Class Venue: CS-11

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 | С | 2 |
| Apply basic programming concepts | С | 3 |
| Design and implement algorithms to solve real world problems and should be able to translate a problem statement into pseudo-code/C++ code | С | 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. Theory and Problems of Programming with C++ by John R. Hubbard, 2nd Edition
- 2. Programming and Problem Solving with C++, Nell Dale
- 3. www.learncpp.com

Tentative Weekly Schedule

| Terrialive weekiy | | 1 |
|---------------------|---|---|
| Week 1 | Lecture 1 | Lecture 2 |
| Problem Solving and | Course introduction and logistics | A brief introduction of programming |
| Programming | | languages and the idea of compiling, |
| Introduction | Introduction to Computers, | linking and loading. |
| | Basic Computer architecture with an | |
| | overview of memory as consisting of | Introduction of some fundamental |
| | addressable storage locations for keeping | operations provided by a Basic/C++ like |
| | data and program. | programming language with/without |
| | | getting into exact C++ program structure |
| | A program as a sequence of instructions and | details. |
| | the Fetch-Decode-Execute cycle | |
| | | |
| | | |
| Week 2 | Lecture 3 | Lecture 4 |
| Simple C++ | The structure of a C++ program with a single | Introduction of logical operations and the |
| Programs | main function and very brief explanation of | use of logical operation for conditional |
| | #include and named spaces. | execution (IF statements). |
| | | |
| | A high level description of some built in C++ | C++ Operators (Arithmetic, Logical, and |
| | datatypes (int, float, double, char, bool), | Relational) Use of operators for different |
| | variables declaration, assignment operator, | datatypes. |
| | input, output. | |
| | | Translating programs written using |
| | | pseudocode or a flowchart into working |
| | | C++code. |
| Week 3 | Lecture 5 | Lacture 6 |
| | | Lecture 6 |
| Simple C++ | Translating programs written using | Programming exercises. |
| Programs | pseudocode or a flowchart into working | Writing clean code using indentation and |
| | C++code continued. | comments. |
| | A basic introduction of operator precedence | |
| | and writing complex expressions as a | |
| | sequence of simple intermediate expressions. | |
| | sequence of simple intermediate expressions. | |
| | | |
| Week 4 | Lecture 7 | Lecture 8 |
| Simple C++ | Using a Nested selection structure. | Programs with nested if/else statements |
| Programs If/Else | Programs with if/else statements | |
| Week F | Lecture 9 | Locking 10 |
| Week 5 | | Lecture 10 |
| Repetition | Repetitions using while, for and do while | Problem solving using repetition structures |
| Structures(Loops) | Lastina 11 | Lastura 13 |
| Week 6 | Lecture 11 | Lecture 12 |
| Nested Control | Problem solving nested repetition structures | Problem solving using nested repetition |
| Structures | Lecture 13 | structures |
| Week 7 Functions | | Lecture 14 |
| FullCuolis | Function definition and calling: | Function Parameters: Pass by value and |
| | parameters and return types; | pass by reference. Stack rolling and |
| | Global and local variables scope and life time. | unrolling. |
| Week 8 | Lecture 15 | Lecture 16 |
| Functions | Top-Down Design of a program and its | Function Overloading. |
| | implementation using functions | Functions with Default Parameters. |
| | Built-in functions | |
| | | |

| Week 9 | Lecture 17 | | | |
|---------------------|---|---|--|--|
| File Handling | I/O from simple text Files | | | |
| Week 9 | Lecture 18 | | | |
| Arrays Introduction | Define and use fixed sized arrays. | | | |
| and Repetition | Array organization in memory and element access using Array name and index. | | | |
| structure | Initialization using member initializer list, and by using loops. | | | |
| | | | | |
| Week 10 | Lecture 19 | Lecture 20 | | |
| Arrays Processing | Printing data, taking input, | Passing arrays to functions. | | |
| | Processing by index and by elements | Design different functions for input, | | |
| | Find Min, Max, Avg, Equilibrium Index | output, search, reverse, | | |
| | Reverse: All Elements, odd/eve elements and | Shifting and Rotation of elements: right | | |
| | indices | and left | | |
| | Search: Linear and Binary | Insert and delete elements from ordered | | |
| March 44 | La store 24 | list using shifting. | | |
| Week 11 | Lecture 21 Passing arrays to functions use of const. | Lecture 22 | | |
| Arrays Processing | Sorting: Bubble Sort, Selection Sort | Sorting: Insertion Sort, Even/odd Sort Merging sorted arrays. | | |
| | Softing. Bubble Soft, Selection Soft | Application: Sets, Union, Intersection, | | |
| | | difference. | | |
| Week 12 | Lecture 23 | Lecture 24 | | |
| CStrings and | I/O from simple text Files in arrays. | Functions design: | | |
| character Arrays | Difference between Null terminated CStrings | Find String length, Compare strings, | | |
| Processing | and character arrays. | Find substring and replace, | | |
| | Storage of CStrings in character arrays and | Calculate frequency of specific characters | | |
| | aggregate I/O. | Remove specific characters. | | |
| Week 13 | Lecture 25 | Lecture 26 | | |
| 2D Arrays | Using built in CSrting functions. | Application: Store and process Students | | |
| Processing | Use of built-in rand () function. | Quiz marks. | | |
| | 2-Dimensional Array and how it is organized | Find Min, Max, Avg, column and row wise. | | |
| | in memory in row/col major order. | Sorting: row wise or column wise, | | |
| | Initialization using member initializer list, and | complete array by specific column or row. | | |
| | by using loops. | | | |
| | I/O and processing of elements in row/col | | | |
| Week 14 | major order. Lecture 27 | Lecture 28 | | |
| 2D Arrays | Passing 2D arrays to functions: Complete, | Application: Matrices storage and | | |
| Processing | individual rows, or elements. | processing | | |
| | Processing diagonals: reverse elements, print | Addition, Subtraction, Multiplication, | | |
| | data of whole array. | Transpose, | | |
| | , | Check for Upper and lower triangular. | | |
| | | Use of graphic libraries functions. | | |
| | | Designing header files for user defined | | |
| | | functions. | | |
| Week 15 | Lecture 29 | | | |
| 2D CStrings | Storage and processing of CStrings in 2D Arrays | 5. | | |
| Processing | Bitwise operators | | | |
| | Binary files I/O | | | |
| | Application: Data compression. | | | |
| Week 16 | Lecture 30 | Lecture 31 | | |
| Structures | Introduction and using structs | Arrays of Structs, | | |
| | Passing and returning from functions | Application of structs | | |
| | | | | |
| Final Exam | | | | |
| | | | | |

(Tentative) Grading Criteria:

1. Assignments + Homework (10 %)

Quizzes (10 %)
 Midterms (25 %)
 Project (10 %)
 Final Exam (45 %)

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

Course Policies:

- All assignments and homework must be done individually.
- o Late Submissions of assignments will not be accepted.
- No re-take of quizzes.
- 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.