|  |  |
| --- | --- |
| Logo Fast | **NATIONAL UNIVERSITY**  **of Computer & Emerging Sciences, Lahore** |

Department of Computer Science

**CS118 – Programming Fundamentals**

**FALL 2020**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor Name:** | Muhammad Waqas Manzoor | | |
| **Email address:** | Waqas.manzoor@nu.edu.pk | | |
| **Office Location/Number:** | Opposite CS-8 | | |
| **Office Hours:** | Tuesday, Wednesday 10:00-11:00 AM | | |
| **TA Information:** |  |  |  |
|  | **Section** | **TA Name:** | **Email address:** |
|  | BDS-1B, 1D | Rana Abdul Muneem | [l191046@lhr.nu.edu.pk](mailto:l191046@lhr.nu.edu.pk) |
|  |
|  |

**Course Information**

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

**Class Meeting Time: Section Timings**

**BDS-1B Tue-Wed 11:30-12:50**

**BDS-1D Thu-Sat 01:00-02:20**

**Class Venue: CE1**

**Course Description/Objectives/Goals:**

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

**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: From Problem Analysis To Program Design, by D. S. Malik (7th Edition)
2. C++: How to Program? by Deitel & Deitel (8th Edition)

**Additional references and books related to the course:**

1. Programming Principles and Practice Using C++ by Bjarne Stroustrup
2. Theory and Problems of Programming with C++ by John R. Hubbard, 2nd Edition
3. Programming and Problem Solving with C++, Nell Dale
4. <https://www.hackerrank.com/>

**Tentative Weekly Schedule [First 8 Sessions]**

|  |  |  |
| --- | --- | --- |
| **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 |
| **MID-I** | | |
| **Week 5**  Repetition Structures(Loops) | **Lecture 9**  Repetitions using while, for and do while | **Lecture 10**  Problem solving using repetition structures |
| **Week 6**  Nested Control Structures  Introduction to Function | **Lecture 11**  Problem solving repetition structures  I/O from simple text Files and C++ | **Lecture 12**  Introduction to creating function  Function calling, parameter passing and return types;  Built-in functions |
| **Week 7**  **Arrays and Repetition Structures** | **Lecture 13**  Define and use fixed sized arrays**.**  Array organization in memory and element access using Array name and index  Problems involving array I/O and accessing elements of an array in various orders | **Lecture 14**  Commonly Performed Operations on 1D-Array like of sorting and searching etc.  **CStrings** |
| **Week 8**  **Functions** | **Lecture 15**  2-Dimensional Array and how it is organized in memory in row/col major order.  I/O elements of a 2D array. | **Lecture 16**  Problems involving 2D array  Multidimensional arrays |
| **Week 9**  **Functions** | **Lecture 17**  Call by value vs call by reference  Passing arrays to functions. | **Lecture 18**  Problems involving Functions |
| **MID-II** | | |
| **Week 10**  **Structures and Unions** | **Lecture 19**  Function overloading  Default parameter values | **Lecture 20**  Static variables and variable scope |
| **Week 11**  **File Handling** | **Lecture 21**  User Defined Data types  Introduction and using structs and unions | **Lecture 22**  I/O from ASCII/Unicode Text Files  I/O from binary files |
| **Week 12**  **Using Built-in classes** | **Lecture 23**  CStrings vs string | **Lecture 24**  Using vectors and Built-in Data structures |
| **Week 13**  **Review** | **Lecture 25**  Using vectors and Built-in Data structures | **Lecture 26**  Review |
| **Final Exam** | | |

**(Tentative) Grading Criteria:**

1. Assignments + Homework **(15 %)**
2. Quizzes **(0 %)**
3. Midterms **(25 %)**
4. Project **(15 %)**
5. Final Exam **(45 %)**
   * 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.