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

Department of Computer Science

**CS118 – Programming Fundamentals**

**FALL 2020**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor Name:** | Waqas Manzoor | **TA Name:** | Muhammad Kamraan |
| **Email address:** | Waqas.manzoor@nu.edu.pk | **Email address:** | [l174308@lhr.nu.edu.pk](mailto:l174308@lhr.nu.edu.pk) |
| **Office Location/Number:** | Liberty Lab |  |  |
| **Office Hours:** | **Thu- Sat 3:00-4:00 p.m.** |  |  |

**Course Information**

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

**Class Meeting Time:** Section E: Thu- Sat 11:00 – 12:30 PM

**Class Venue: E&M-11**

**Class Meeting Time:** Section F: Thu- Sat 12:30 – 02:00 PM

**Class Venue: E&M-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 | 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 Deitle & Deitle (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**

|  |  |  |
| --- | --- | --- |
| **Week 1**  Problem Solving and Programming  Introduction | **Lecture 1**  Introduction to Computing,  Basic Computer architecture, Computational Problems, Algorithms and Problem Solving | **Lecture 2**  Thinking logically, Solving logical problems.  Programming Language, Program Compiler and Linker, Basic output, Concept of variables. |
| **Week 2**  Simple C++ Programs | **Lecture 1**  C++ simple datatypes (int, float, double, char, bool), variables declaration, assignment operator, input, output. | **Lecture 2**  C++ Operators (Arithmetic, Logical, and Relational)  Use of operators for different datatypes. |
| **Week 3**  Simple C++ Programs | **Lecture 1**  Precedence of Operators.  Designing complex expressions for mathematical type conversion problems. Typecasting in C++. | **Lecture 2**  Logical decision making and solving logical problems with selection structure. |
| **Week 4**  Simple C++ Programs If/Else | **Lecture 1**  Programs with if/else statements | **Lecture 2**  Programs with nested if/else statements |
| **MID 1** | | |
| **Week 6**  Loops | **Lecture 1**  Repetitions using while loops | **Lecture 2**  Problems on while loops |
| **Week 7**  Nested Loops | **Lecture 1**  Repetitions using while and for loops | **Lecture 2**  Problems on nested loops |
| **Week 7**  Nested Loops and If/Else | **Lecture 1**  Repetitions with selection structure | **Lecture 1**  Nested loops with selection structure |
| **Week 8**  **Functions** | **Lecture 1**  Function definition and Calling:  Parameters and return types; | **Lecture 2**  Types of parameters (value and reference) stack rolling and unrolling**.** |
| **Week 9**  **Functions**  & File I/O Collections of data | **Lecture 1**  Reading and Writing from text files | **Lecture 2**  How to define and use fixed sized arrays**.**  Arrays, if/else and loops |
| **Week 10**  CStrings | **Lecture 1**  Linear and Binary Search, Array Reversal, Detecting Palindromes | **Lecture 2**  Character arrays and CStrings |
| **MID 2** | | |
| **Week 11**  Arrays, functions | **Lecture 1**  Character arrays and CStrings | **Lecture 2**  Functions with 1D array parameters |
| **Week 12**  Functions with nested loops | **Lecture 1**  Bubble Sort & Selection Sort | **Lecture 2**  Substring Finding |
| **Week 13**  Multidimensional arrays | **Lecture 1**  How multidimensional arrays are stored and used | **Lecture 2**  Programs with multidimensional arrays, e.g. matrix addition and multiplication |
| **Week 14**  Char Arrays with functions | **Lecture 1**  2D char arrays | **Lecture 2**  Bitwise Operators |

**(Tentative) Grading Criteria:**

1. Assignments + Homework **(10 %)**
2. Quizzes **(10 %)**
3. Midterms **(25 %)**
4. Project **(10 %)**
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:**

* + Quizzes may be announced or surprise.
  + No Makeup Quizzes.
  + 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.