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

Department of Computer Science

**CS-1004 – Object Oriented Programming**

**Spring 2022**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor Name:** | Samin Iftikhar | **TA Name:** | TBA |
| **Email address:** | Samin.iftikhar@nu.edu.pk | **Email address:** | [TBA](mailto:l201314@lhr.nu.edu.pk) |
| **Office Location:** | Exam Hall, old admin block |  |  |
| **Office Hours:** | **Wed -11:30-12:30** |  |  |

**Course Information**

**Program:** BS (CS)

**Credit Hours:** 3 + (1 for Lab )

**Type:** Core

**Class Venue:** CS-4

**Pre-requisites:** Programming Fundamentals (CS-118)

**Class Meeting Time:** BCS-2E Tue 11:30-13:00, Wed 13:00-14.30, BCS-2F Tue 13:00-14:30, Wed 10:00-11.30

**Course Description/Objectives/Goals:**

The core objectives of this course are to introduce,

* Object oriented programming with data abstraction and encapsulation.
* The classes, objects and relationship among different objects and classes in C++?
* Generic programming using templates, and template specializations.

**Course Learning Outcomes (CLOs):**

|  |  |  |
| --- | --- | --- |
| **At the end of the course students will be able to:** | **Domain** | **BT\* Level** |
| Understand dynamic memory management with pointers. | C | 2 |
| Understand principles of object oriented program | C | 2 |
| Identify the objects & their relationships to build object oriented solution | C | 3 |
| Model a solution for a given problem using object oriented principles | C | 3 |
| Examine an object oriented solution | C | 4 |
| \* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain | | |

**Course Textbooks:**

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. Problem Solving with C++, by Walter Savitch
2. https://www.learncpp.com

**Course Contents Weekly and Lecture-wise Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Topic** | **Lecture-1** | **Lecture-2** |
| 1 | **Pointers** | Pointers Introduction, Pointer variables and Initialization, Address of Operator, Dereferencing Operator. | Pointer Operations (Relational, Arithmetic) |
| 2 | Use of Constant with Pointers.  Difference between a Pointer and a Reference.  Passing pointers to functions by value and by reference. | Dynamic memory allocation using pointers and accessing dynamic memory. Dynamic Variables new and delete operators. |
| 3 | Dynamic 1- dimensional arrays, Create, Delete, Grow and Shrink.  Example of programs using 1D dynamic allocation: e.g., mathematical sets union and intersection. | Memory Leak and Dangling Pointers, Shallow copy Vs. Deep Copy |
| 4 | Partial Dynamic 2D, allocation, | Pointers Multiple Indirection Dynamic 2D Arrays, 3D Arrays. |
| 5 | Pointers Indirection Matrices, and functions. How to pass different levels of pointers in functions using indirection. | Dynamic Character Arrays and CStrings.  Structured Programming vs Object-oriented Programming, Principles of modularization, abstraction and encapsulation.  Objects vs Class, state vs behavior, access specifiers (Public, Private), Member functions (accessors, utilities, mutators etc) |
| **6** | **Mid Term 1** | | |
| 7 | **Object-oriented basics** | Constructors (default, overloaded), Function overloading. | Dynamic memory allocation and Object assignment, Parameter passing, |
| 8 | Shallow vs Deep copy, Copy constructor, Destructors, this pointer, | Cascaded function calls, static members, inline functions and other miscellaneous issues. |
| 9 | **Operator overloading** | Unary operators using member functions | Binary operators using member functions |
| 10 | Binary operators using non-member functions, concept of friendship, | Unary operators, Pre and post increment, subscript operator. |
| 11 | **Object and Class relationships** | Part-whole relationships, Association/Aggregation | Composition  Implementation issues (constructor call sequence, initializer list, etc) |
| **12** | **Mid Term 2** | | |
| 13 |  | Inheritance basics, Type of Inheritance, public, protected, private. | Function Overriding and sub-typing details |
| 14 | **Object and Class relationships** | Polymorphism introduction Static vs dynamic binding details, virtual tables and virtual pointers, | Polymorphism vs down casting, run-time type identification, dynamic cast |
| 15 | Pure-virtual functions, Abstract classes, Interfaces (optional) | Multiple Inheritance and Diamond Problem Multiplicity, Memory Management  Bi-directional relationships, Forward-class declarations issues |
| 16 | **Generic Programming**  **&**  **Exception Handling.** | Template functions | Template classes  Template Specializations, |
| Exception Handling. | Introduction to STL, Iterators and Collections |

**(Tentative) Grading Criteria:**

1. Assignments + Project **(20 %)**
2. Quizzes **(10 %)**
3. Midterms **(30 %)**
4. Final Exam **(40 %)**
   * Grading scheme for this course is **Absolute** under application of FSC'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 to appear in the Final exams.