**NATIONAL UNIVERSITY**
of Computer & Emerging Sciences, **Lahore**

# Department of Computer Science

## CS-217 – Object Oriented Programming
## Spring 2023

| | | | |
|---|---|---|---|
| **Instructor Name:** | Mubashar Hussain | **TA Name:** | |
| **Email address:** | Mubashar.hussain@nu.edu.pk | **Email address:** | |
| **Office Location:** | **M-126 Old Exam Hall** | | |
| **Office Hours:** | **Tue-Thu -01:00-3:30 p.m.** | | |

## Course Information

**Program:** BS (CS)

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

**Type:** Core

**Class Venue:** CS - 3

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

## 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: |
|---|
| **Demonstrate** the basic concepts of OOP |
| **Apply** OOP concepts (Encapsulation, Inheritance, Polymorphism, Abstraction) to computing problems for the related program |
| **Model** an algorithmic solution for a given problem using OOP |
| **Apply** good programming practices |

## 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:**

3. Problem Solving with C++, by Walter Savitch
4. 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) | Use of Constant with Pointers. Difference between a Pointer and a Reference. Passing pointers to functions by value and by reference. |
| 2 | | Dynamic memory allocation using pointers and accessing dynamic memory. Dynamic Variables new and delete operators. | Dynamic 1- dimensional arrays, Create, Delete, Grow and Shrink. Example of programs using 1D dynamic allocation: e.g., mathematical sets union and intersection. |
| 3 | | Memory Leak and Dangling Pointers, Dynamic 1- dimensional char arrays for strings, string operations like search, concatenation etc. | Pointers Indirection. Dynamic 2D, allocation, matrices, CStrings etc. |
| 4 | **Object-oriented basics** | 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) |
| 5 | | Constructors (default, overloaded), Function overloading. | Dynamic memory allocation and Object assignment, Parameter passing, Shallow vs Deep copy, |
| 6 | **Mid Term 1** | | |
| 7 | **Object-oriented basics** | Copy constructor, Destructors, this pointer, | Cascaded function calls, static members, inline functions and other miscellaneous issues |
| 8 | **Operator overloading** | Unary operators using member functions | Binary operators using member functions |
| 9 | | Binary operators using non-member functions, concept of friendship, | Unary operators, Pre and post increment, subscript operator. |
| 10 | **Object and Class relationships** | Part-whole relationships, Association/Aggregation | Composition Implementation issues (constructor call sequence, initializer list, etc) |
| 11 | | Inheritance basics, Type of Inheritance, public, protected, private. | Function Overriding and sub-typing details |
| 12 | **Mid Term 2** | | |
| 13 | **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 |
| 14 | | Pure-virtual functions, Abstract classes, Interfaces (optional) | Multiple Inheritance and Diamond Problem Multiplicity, Memory Management Bi-directional relationships, Forward-class declarations issues |
| 15 | **Generic Programming & Exception Handling.** | Template functions | Template classes Template Specializations, |
| 16 | | Exception Handling. | |

## (Tentative) Grading Criteria:

1. Assignments + Home works + Project **(15 %)**
2. Quizzes                                      **(10 %)**
3. Midterms                               **(30 %)**
4. 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:

o All assignments and homework must be done individually.

o Late Submissions of assignments will not be accepted.

o No retake of announced quizzes

o **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.

o Minimum **80%** attendance is required for appearing in the Final exams.