

OBJECT ORIENTED PROGRAMMING (THEORY) CS-101

Pre-requisite: Fundamentals of Programming

Credit Hours 01

Contact Hours 16

TEXT BOOKS

- Object-Oriented Programming in C++, 4th edition, by Robert Lafore.

REFERENCE BOOKS

- Schaum's Outlines Programming with C++, 2nd edition by John R. Hubbard
- C++ How to Program, 8th edition, By Dietel & Dietel.

COURSE OBJECTIVES

The objective of this course is to make students learn about fundamental concepts of Object Oriented Programming and its implementation in C++ language. The course covers the concepts of classes, objects, attributes, operator overloading, inheritance and virtual functions.

S. NO.	CLO/ PLO MAPPING	DOMAIN	PLO
1.	Define and identify fundamental concepts of object-oriented programming.	C1	01
2.	Apply principles of object oriented programming to a particular situation.	C3	02
3.	Design and develop small-to-moderate programs to manipulate and manage data elements while exhibiting the object-oriented programming skills	C5	03

COURSE CONTENTS

Objects and Classes

- Introduction
- Class, objects, attributes and members functions
- The Unified Modeling Language (UML).
- Constructors and destructors, constructor overloading,

- Defining member functions outside the class. objects as function arguments. The default copy constructor. Returning objects from a function.
- Structures, classes, objects and memory. static class data.
- const member functions and const Objects.

Operator Overloading

- Introduction.
- Overloading unary (++ , -- , ! , ~) operators.
- Operator argument and return values, postfix and prefix unary operator overloading.
- Returning nameless temporary objects
- Binary operator overloading(+, -, * , /), adding two objects, concatenating two String objects Multiple Overloading: Overloading comparison (<, ==) and arithmetic assignment (+=) operators.
- Overloading [] operator using returning by reference.
- Conversions between objects and basic types(int, float), conversion between C-Strings (char*) and String objects.
- Conversions between objects of different classes, 2D Polar to Cartesian coordinates conversion.
- Preventing conversions with explicit, Changing const object data using mutable.

Inheritance

- introduction, concept of specialization and generalization, Derived class and base class. UML Diagrams. Derived class constructors.
- Overriding member function examples. Using Inheritance in a database example.
- Level of inheritance, Multiple Inheritance, Member functions in Multiple Inheritance. Constructors in Multiple Inheritance. Ambiguity in multiple inheritance.
- Aggregation: Classes within classes, Using aggregation in programming. Composition: a stronger aggregation

Virtual Functions

- Normal and Virtual Member Functions Accessed with Pointers.
- Abstract Class and pure virtual functions.
- Virtual Destructors, Virtual Base Classes.
- Friend Functions and Friend Classes, Static Functions.