Lab no :08

Lab name: Introduction to Object-Oriented Programming (OOP) in C++

Introduction: Object-Oriented Programming (OOP) is a programming paradigm that revolves around the concept of objects, which are instances of classes. It allows for the organization of code into manageable units, promotes code reusability, and enhances code maintainability. C++ is a powerful programming language that supports OOP principles and provides features such as classes, inheritance, polymorphism, and encapsulation.

Objective: The purpose of this lab is to introduce the basics of object-oriented programming (OOP) languages, including key concepts such as classes, objects, inheritance, encapsulation, and polymorphism.

Equipment:

* Computer
* CodeBlocks Software

Theory:

* Classes and Objects:
  + - A class is a blueprint for creating objects. It defines the properties (data members) and behaviors (member functions) of objects.
    - An object is an instance of a class. It encapsulates data and behaviors defined by its class.
* Inheritance:
  + - Inheritance allows a class (subclass/derived class) to inherit properties and behaviors from another class (superclass/base class).
    - It promotes code reuse and establishes a hierarchical relationship between classes.
* Polymorphism:
  + - Polymorphism allows objects of different classes to be treated as objects of a common superclass.
    - It enables methods to behave differently based on the object they are invoked upon, leading to code flexibility and extensibility.
* Encapsulation:
  + - Encapsulation refers to the bundling of data and methods that operate on the data within a single unit (class).
    - It restricts access to the data from outside the class and only allows access through specified methods, thus enhancing data security and modularity.

Benefits of OOP in C++

* Code Reusability: Inheritance allows for the creation of class hierarchies, where derived classes reuse code from base classes.
* Maintainability: Encapsulation promotes modularity and makes code easier to understand and modify.
* Improved Code Organization: OOP encourages a more structured approach to code organization by grouping data and functionality together.
* Real-world Modeling: OOP provides a natural way to model real-world entities and their relationships through objects and classes.

Discussion In this lab, we explored the basics of Object-Oriented Programming in C++ by studying classes, objects, inheritance, polymorphism, and encapsulation. Through a simple example, we demonstrated how these concepts can be implemented in C++ to create modular, reusable, and maintainable code. Understanding OOP principles is essential for developing complex software systems and applications