

## Experiment No - 3

Page No.:

Date

youva

19UC5122

Title: Write a program to implement a Date class.  
Objectives:

1. To understand the concept of abstraction in C++.
2. To understand the concept of encapsulation in C++.

Key Concepts: Access modifiers, accessor, mutator, abstraction, encapsulation.

Theory:

Access Modifiers: There are 3 types of access modifiers available in C++: (1) Public

(2) Private

(3) Protected

Public:

All the class members declared under public will be available to everyone. The data members and member functions declared public can be accessed by other classes too. The public members of a class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

Private:

The class members declared as private can be accessed only by the functions inside the class. They are not allowed to be accessed directly by any object or function outside the class. Only the member functions or the friend functions are allowed to access the private data members of a class.

Protected: Protected access modifiers is similar to that of private access modifiers, the difference is that the class member declared as protected are inaccessible outside the class but they can be



19UCS122

accessed by any subclass (derived class) of that class.

Mutator  
Accessor function: A member function of a class that modifies the value(s) of the member variable(s).

Accessor function: A member function of a class that only accesses (that is, does not modify values) of the member variables.

Because an accessor function only accesses the values of the member variables, as a safeguard we typically include the reserved word `const` at the end of the headings of these functions. A member function of a class is called a constant function if its heading contains the reserved word `const` at the end.

Encapsulation (Data binding & data hiding)

Binding of data and code together is called "class is a basic unit of encapsulation". Gathering the implementation details together data binding and separating them from the abstraction is called encapsulation. Data hiding (putting data into the private section of a class) is an instance of encapsulation, and so is hiding functional details of an implementation in the private section. Encapsulating data in the private section protects the integrity of data and is called data hiding. Thus using a class is the C++ way of making it easy to implement the oop features abstraction, data hiding and encapsulation.

Abstraction and classes

Life is full of complexities, and one way we cope with complexity is to frame simplifying abstraction. You are a collection of more than an octillion atoms. Some



19UCS122

Page No.:

Date:

youva

students of the mind would you say that your mind is a collection of semiautonomous agents. But it's much simpler to think of yourself as a single entity.

In computing, abstraction is the crucial step of representing information in terms of its interface with the user. That is, you abstract the essential operational features of a problem and express a solution in those terms. In the softball statistics example, the interface describes how the user initializes, updates and displays the data. From abstraction, it is a short step to the user-defined type, which in C++ is a class design that implements the abstract interface.

Abstraction for the car we drive, most of us want to know how to start the car and drive it. Most people are not concerned with the complexity of how the engine works. In other words abstraction focuses on what the engine does and not on how it works abstraction is the process of separating the logical properties from the implementation details.