Experiment No-04.

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++.

Theory:

Access modifiers:-

There are three 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 class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

Protected:

Protected access modifier is similar to that of private access modifier, the difference is that the class member declared as protected are inaccessible outside the class but can be accessed by subclass.

Private:

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

Accessor function:-

A member function of a class that only access (that is, does not modify the value(s) of member variable(s).

Mutator function:-

A member function of class that modifies the value(s) of member variable(s).

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

Encapsulation (Data Binding and 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 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 the data and is called data hiding. Thus using a class is the c++ way of making it easy to implement the OPP 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 abstractions. You are a collection of more than an octillion atom. Some students of the mind would 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 short step to the user-defined type, which in c++ is a class design that implements the abstract interface.

Abstraction for the car that 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 how it works. Abstraction is the process of separating the logically properties from implementation details.

Example in Date class, the actual implementation of the operations, that is, the definations of the member functions of the class postponed. So, a data type that separates the logical properties from the implementation details known as abstract Data Type (ADT).

Is it a good practice to include the specification and implementation details of class in the program? Definitely not. Suppose the definition of the class and the definition of the member function are directly included in the user's program. The user then has direct access to the definition and the definition of the member function. Therefore user can modify the operation in any way the user pleases. Thus we must hide the implementation details. Hiding the implementation details frees the user from having to fit the extra piece of code in the program. Furthermore once an object has been written, debugged and tested properly, it becomes (and remains) error free.

Problem statement:-

write a Date class having following attributes and functions.

Problem Analysis:-

To implement Data in a program, the user must declare object of type Date and know which operation are allowed and what the operations do. So the user must have access to the specification details. Because of the user is not concerned with the implementation details, we must put those details in a separate file called an implentation file. Also, because the specification details can be too long. Therefore we must put the specification details in a separate file. The file that contains the specification details is called the header file (or interface file).

Properties (Member variables)
(·) day, month, year - declare all variable private.

Member functions:-
(·) Constructors: supply following constructors
(·) No argument constructor: Date()
(·) Overloaded constructor: Date(int d, int m, int y).
(·) Setters and getters for member variable
(·) increment date - increment current daty by 1
(·) decrement date - decrement current date by 1.
(·) date differen -
call the function and second date as argument).