EXPERIMENT NO-1
Title: Write a program to implement class in C++.

Objectives:
1. To understand the concept of class and object.
2. To understand the concept of class members and the access specifiers.

Theory:
In Object oriented design (OOD), the first step is to identify the components, called objects. An object combines data and the operations on that data in a single unit. In C++, the mechanism that allows you to combine data and the operations on that data in a single unit is called a class. A class is a collection of a fixed number of components. The components of a class are called the members of the class.
The general syntax for defining a class is: class

```
class classidentifier
{
        Classmemberlist;
};
```

in which classMembersList consists of variable declarations and/or functions. That is, a member of a class can be either a variable (to store data) or a function.

- If a member of a class is a variable, you declare it just like any other variable. Also, in the definition of the class, you cannot initialize a variable when you declare it.
- If a member of a class is a function, you typically use the function prototype to declare that member.

The members of a class are classified into three categories: private, public, and protected. In C++, private, protected, and public are reserved words and are called member access specifiers.
Following are some facts about public and private members of a class:

- By default, all members of a class are private.
- If a member of a class is private, you cannot access it outside of the class.
- A public member is accessible outside of the class.
- To make a member of a class public, you use the member access specifier public with a colon:

Accessing Class Members

Once an object of a class is declared, it can access the members of the class. The general syntax for an object to access a member of a class is:

classObjectName.memberName

The class members that a class object can access depend on where the object is declared.

• If the object is declared in the definition of a member function of the class, then the object can access both the public and private members.

• If the object is declared elsewhere (for example, in a user's program), then the object can access only the public members of the class.

In C++, the dot, (period), is an operator called the member access operator.

Class Scope

A class object can be either automatic (that is, created each time the control reaches its declaration and destroyed when the control exits the surrounding block) or static (that is, created once, when the control reaches its declaration, and destroyed when the program terminates). Also, you can declare an array of class objects. A class object has the same scope as other variables. A member of a class has the same scope as a member of a struct. That is, a member of a class is local to the class. You access a class member outside of the class by using the class object name and the member access operator (.).

Functions and Classes

The following rules describe the relationship between functions and classes:

- Class objects can be passed as parameters to functions and returned as function values.
- As parameters to functions, class objects can be passed either by value or by reference.
- If a class object is passed by value, the contents of the member variables of the actual parameter are copied into the corresponding member variables of the formal parameter.

Constructors and Destructors:

Constructors and destructors are two special kinds of member functions. In general, a constructor is a member function with the same name as the class. A constructor is the first method that is called implicitly when an object is created. They are used to initialize data and provide the guarantee that the data is always valid.

A destructor is the method that is called each time the object dies or exceeds its lifetime. It has the same name as the name of the class, prefixed with a ~ (character tilde). They are used to perform any cleanup activity for data members whose memory is allocated dynamically.

For the above example

myClassA() ; //constructor

~myClassA(); //destructor constructors and destructors need to have public scope.

Constructors can be overloaded, to support different ways of object initialization.

Procedure:

1. Declare a class Person with following members:

    Properties (Member variables):

    name,address,phone declare as private.

    Member Functions:

    setName, getName,setAddress,getAddress,setPhone,getPhone

    no argument and parameterised constructors.
2. Create an object of Person class
3. Assign values to the members of the person object using set methods.
4. Read and display the values of member variables of Person object using get method.

Keywords:

class, object, constructor, destructor, member variables and functions, access specifiers.