

Experiment-03

PAGE NO.:

DATE: / /

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 constructor and destructor.

Key Concepts: class, object, constructor, destructor.

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 classidentification
{
    class member list;
    :
};
```

in which classMemberList consists of variable declarations and/or functions. That is, a member of a class can be either a variable or function.

- (*) If a member of a class is a variable, you declare it just like any other variable. Also, in the definition of 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 class.
- (*) A public member is accessible outside of class.
- (*) To make a member of a class public, you use the member access specifier public with colon:

Accessing class members

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

classObjectName.memberName.

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

- (i) If the object is declared in the definition of a member function of the class, then the object can access both public & private members.
- (ii) If the object is declared elsewhere, then the object can access only the public members of class.

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

Class Scope:-

A class object can be either automatic or static 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 class has same scope as 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 (\cdot).

Functions and classes:-

The following rules describes the relationship between functions and classes:-

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

Constructor and destructor:

Constructor and destructor are two special kinds of member functions. In general, a constructor is a member function with the same name as that class. A

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 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 that are allocated dynamically.

For the above example

```
myclassA(); // constructor  
~myclassA(); // destructor.
```

Construction can be overloaded to support different ways of object initialization.

Problem Statement:

1. Write address class with following attributes and functions.
2. A person class having following attributes and functions.
3. A Product class having following attributes and functions.

Problem Analysis:-

1. For address class:-

Properties

(private) street1, street2, city, pin, district, state, country.

Member functions:-

(i) No argument constructor: Address()

(ii) initialize Address

(iii) Display.

2. For Person class:

Member variables:

name, Address (type class), email, mobile, birthdate

Member functions:-

constructor

non parameterized & parameterized

accept person Details()

display person Details()

3. For Product class:

Member variables:-

productid, Product Name, purchase-price, sale price, discount, stock quantity.

Member functions

constructor

non parameterized

initialize product()

display product info()