Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**7**
Student Name:	**Deep patel**
Roll No :	**8**

Title:

1. To write a program to demonstrate friend function in C++.
2. To write a program to demonstrate friend class in C++.

Learning Objective:

- Students will be able to implement friend function and friend classes in C++.

Learning Outcome:

- To understand how to use the private members using friend function and friend class.

Course Outcome:

ECL304.6	Percept the Utility and applicability of OOP

Theory:

1. Explain in details about access specifiers: public, private and protected.

In C++, there are three access specifiers:

Public Class - members are accessible from outside the class, f public access specifier is used while deriving class then the public data members of the base class become the public member of the derived class and protected members becomes the protected in the derived class but the private members of the base class are inaccessible.

Private Class - members cannot be accessed (or viewed) from outside the class, If private access specifier is used while creating a class, then the public and protected data members of the base class become the private member of the derived class and private member of base class remains private.

Protected Class - members cannot be accessed from outside the class, however, they can be accessed in inherited classes. You will learn more about Inheritance later. If protected access specifier is used while deriving class, then the public and protected data members of the base class becomes the protected member of the derived class and private member of the base class are inaccessible.

2. Explain about friend function and friend classes in C++.

a. Friend Class: A friend class can access private and protected members of other class in which it is declared as friend. It is sometimes useful to allow a particular class to access private members of other class.

b. Friend Function Like friend class, a friend function can be given a special grant to acess private and protected members. A friend function can be:

   a) A member of another class

   b) A global function

Program 1: To write a program to demonstrate friend function in C++.

Algorithm:

STEP 1: Start

STEP 2: Create class office

STEP 3: Define attributes and friend function

STEP 4: Mention friend function and its attributes

STEP 5: Create object in main function

STEP 6: Display output

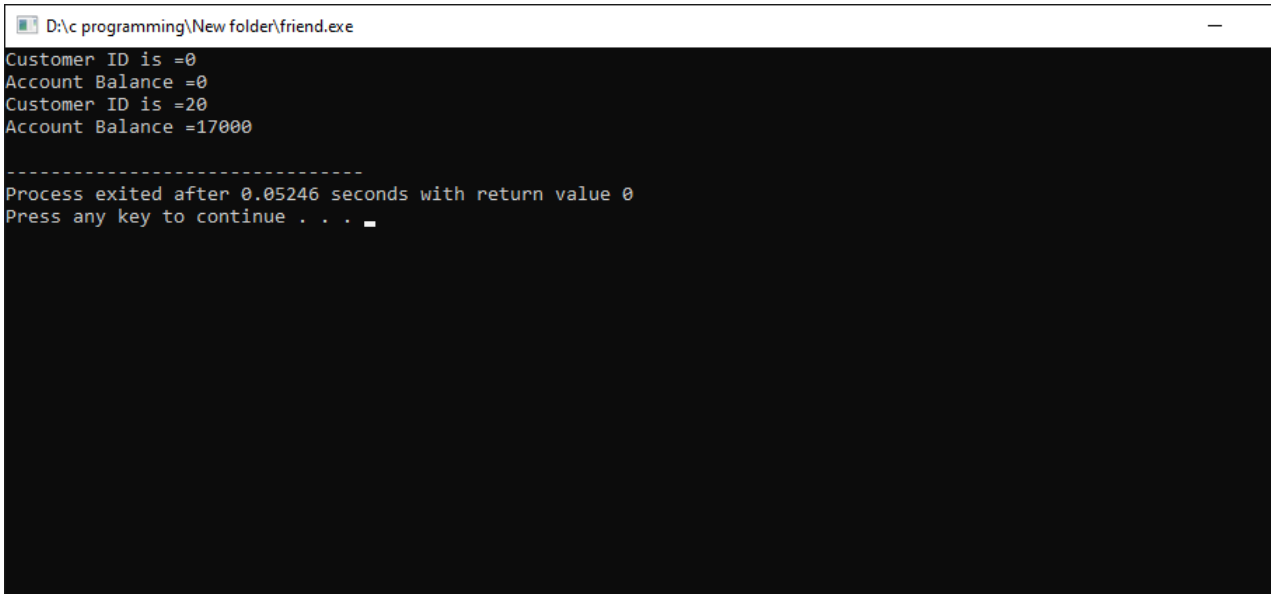
STEP 7: Stop

Code:

```
#include<iostream>
using namespace std;

class Office
{
        int empID;
        float Salary;

        public:
        Office(){
                empID=0;
                Salary=0;
        }

        void displayDetails(){
```

```
                cout<<"Customer ID is ="<<empID<<endl<<"Account Balance
="<<Salary<<endl;
        }
        friend void insertDetails(Office &obj);
};
void insertDetails(Office &obj){
                obj.empID=20;
                obj.Salary=17000;
}

int main(){
        Office obj;
        obj.displayDetails();
        insertDetails(obj);
        obj.displayDetails();
        return 0;
}
```

OUTPUT:



Program 2: To write a program to demonstrate friend class in C++.

Algorithm:

Step 1: start

Step 2: create class one

Step 3: define attributes and mention friend class 'class two'

Step 4: create class two

Step 5: Define attributes and method void displayone(one & a)

Step 6: Create object in main function

Step 7: Display output

Step 8: Stop

Code:

```
#include <iostream>
class one {
private:
        int x;

public:
        one() { x = 8 ; }

        friend class two;
};

class two {
private:
        int y;
```

```
public:
        void displayone (one & a)
        {
                std::cout << "one 'x' = " << a.x;
        }
};


int main()
{
        one x;
        two y;
        y.displayone(x);
        return 0;
}
```

Output:

```
one 'x' = 8
--------------------------------
Process exited after 0.05275 seconds with return value 0
Press any key to continue . . .
```