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:	**Shreyas Sanjay Nanaware**
Roll No :	**39**

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:

- Explain in details about access specifiers: public, private and protected.
- Explain about friend function and friend classes in C++.

Ans:

Access Modifiers or Access Specifiers in a class are used to assign the accessibility to the class members. That is, it sets some restrictions on the class members not to get directly accessed by the outside functions.

1. Access Specifiers define how members (attributes and methods) of a class can be accessed.
2. Below listed are the access specifiers in C++: -

3. **Public:** Members can be accessible from outside the class.
4. **Private:** Members cannot be accessed (or viewed) from outside of the class.
5. **Protected:** Members cannot be accessed from outside of the class but however, they can be accessed in inherited classes.
6. By default, all members of a class are private if you don't specify an access specifier.

Friend class:

A friend class is a class that can access the private and protected members of a class in which it is declared as friend. This is needed when we want to allow a particular class to access the private and protected members of a 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.

For understanding the basic syntax , we have the following program:

```
class A
{
   friend class B;

};

class B
{

};
```

Class B is declared as a friend of class A in the above code. So, now all the member functions of class B became friend functions of class A.

Friend function:

If a function is defined as a friend function in C++, then the protected and private data of a class can be accessed using the function.

By using the keyword friend compiler knows the given function is a friend function.

For accessing the data, the declaration of a friend function should be done inside the body of a class starting with the keyword friend.

Now for understanding the syntax we will take a simple example:

```
class first_class
{
   int  a;
   public:
      first_class()
      {
         a= 0;
      }
      friend int second_class ( first_class)     //declaring friend function
};
```

So in the above example , we have declared a class named first_class and inside it we have a private attribute int a. Then we have made a constructor inside of which we have initialized the value of the variable as zero.

In the same class only we declare the statement of the friend function using the friend keyword as the prefix for the class name and its type.

But the body of the friend function is always outside the class.

Difference between the friend class and the friend function:

Friend function is a function that is able to access the private and protected members of a class. In contrast, a friend class is a class which help in accessing the private members of a class.

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

Algorithm:

Step 1: Create a class 'class1' and inside that class put the various attributes of the class (in private), calculation method for calculating the area of the square and rectangle and also declare the friend function inside this class.

Step 2: The body of the friend function is outside of the class1.

Step 3: Using the friend function we will access the private data attributes of the class and will input the values of length and breadth as 10 using call by reference method.

Step 4: In the main function pass the object a in the method input values.

Step 5: In the main function create the object of the class1 and using that object call the calculation method for getting the output.

Program:

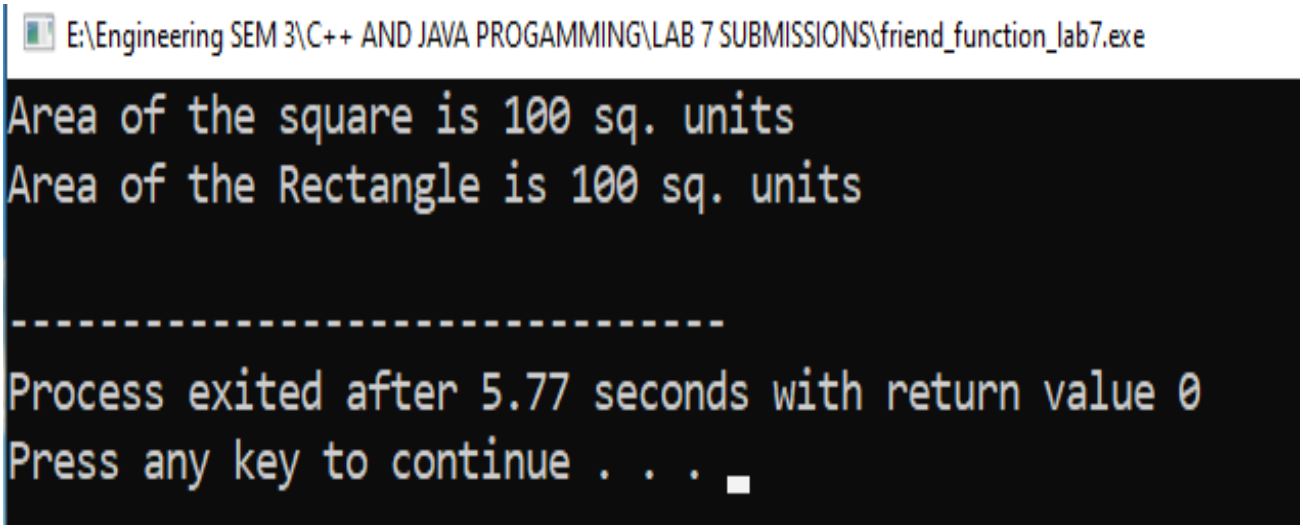
```
/* To write a program to demonstrate friend function in C++.  */


#include <iostream>
using namespace std;
class class1{  //creating a class
  int length , breadth ;  //attributes
  float square_area , rectangle_area; //attributes
  public:
      class1(){ //constructor

          length = 0 , breadth = 0 ;   //giving the initial values of the variables
```

```
        }


        void calculation(){  //calculation part and ouput statement
            square_area=length*breadth;
            rectangle_area=length*breadth;
    cout<<"Area of the square is "<<square_area<<" sq.
units"<<"\n"<<"Area of the Rectangle is "<<rectangle_area<<" sq.
units"<<"\n";
}

friend void input_values(class1 &a);  //friend function


};
void input_values(class1 &a){  //body of the friend function outside the class
    a.length=10;
    a.breadth=10;
}
int main()
{
    class1 a;  //creating the object of the class
    input_values(a);
    a.calculation();  //calling the method using the object
    return 0;
}
```

Input given:

Length=10 , Breadth=10

Output:



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

Algorithm:

Step 1: Create a class named first_class and declare its attributes in private form as int A and int B and initialize them for values 100 and 200 respectively.

Step 2: Create different variables, each representing a mathematical operation such as multiplication , division, addition and subtraction of the two numbers A and B.

Step 3: Creating the friend class of the first_class name ‘second_class’ and declaring its body outside of the class.

Step 4: The body of the friend class include the display_output method through which the output is to be displayed using the object of the class first_class.

Step 5: In the main function we create the objects of both the classes and we call the method for getting the output.

Program:

```
/* To write a program to demonstrate friend class in C++.  */

#include <iostream>
using namespace std;
class first_class{          //creating a class
private:              //access specifier
  int A = 100;   //attributes
  int B = 200;
  int C = A+B;
  int D = A-B;
  int E = A*B;
  float F = (float)A/B ;

public:               //access specifier
  friend class second_class;  //declaring the friend class
};

class second_class {
public:  //access specifier
  void display_output(first_class object1){

    cout<<"Value of A is "<<object1.A<<endl;   //performinng various
mathematical operations sucha as multiplication , addition, division ,subtraction
    cout<<"Value of B is "<<object1.B<<endl;
    cout<<"Value of C= A+B is "<<object1.C<<endl;
```

```
    cout<<"Value of D= A-B is "<<object1.D<<endl;
    cout<<"Value of E= A*B is "<<object1.E<<endl;
    cout<<"Value of F= A/B is "<<object1.F<<endl;
  }
};
int main() {
  second_class object1; //object creation
  first_class object2;
  object1.display_output(object2);  //calling the method using the object of the class
  return 0;
}
```

Input given:

A=100 and B=200

Output: