Lab Number:	7
Student Name:	**Sakshi Vadiraj Kaveri**
Roll No :	**33**

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.

   Data hiding is a very important aspect of object-oriented programming. Data hiding means data is hidden from the outside world. The members that have been declared private are only accessible by the methods declared in that respective class only. The private members of the class are hidden from the outside world. The access specifiers decide the scope and visibility of the data members and functions. There are three access specifiers in C++. They are: public, private, protected.

→ Private access specifier: If you don't specify any access modifier to the data members or member functions of the class then they are by default private. They are hidden from the outside world and they are used to implement data hiding concept of OOP. But only member functions and friends of the class can use them in which they are declared.

→ Public access specifier: All data members and member functions declared using public access specifier are accessible by any function in the program.

→ Protected access specifier: These can only be used by member functions and friend of the class as well as derived class member functions. They are similar to private members since they cannot be accessed directly by non-member function but can be used by derived ones.

Faculty: Ms. Deepali Kayande

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

C++ allows us to make common function friendly to both the classes thereby allowing the function to have access to the private data also the function need not to be a member function of the class. Function declaration uses the keyword 'friend' but function definition need not. Also, function can be declared to be a friend of any number of classes if 2 or more classes are not related then friend function acts like a bridge between them. It can have object as arguments, it cannot access the member functions directly and has to use an object name and operator (.) with each member name. Friend classes are similar to friend function but just the difference is we are making an entire class as a friend class of the main class so that the friend class can have access to the private data members and all functions of the class.

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

ALGORITHM:
1) Declare variables a and b in class X. Assign a=10, b=20 in the method setData().
2) Declare friend function calc() in the class X.
3) In the function calc() take arguments as object of class X and return a+b/2.
4) Display the output.

PROGRAM:
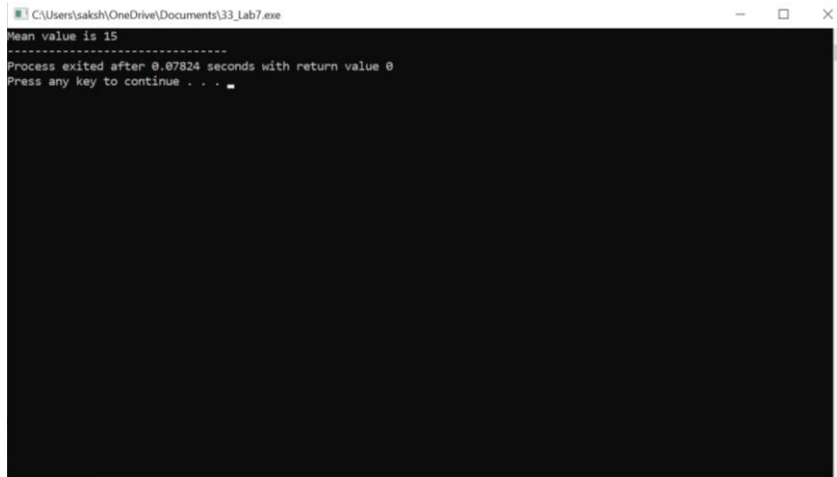
```
#include<iostream>
using namespace std;
class X{
        int a;
        int b;
        public:
                void setData(){
                        a=10;
                        b=20;
                }
                friend int calc(X &o1);
};
int calc(X &o1){
        return int(o1.a+o1.b)/2.0;
}
int main()
{
        X obj;
        obj.setData();
        calc(obj);
        cout<<"Mean value is "<<calc(obj);
        return 0;
}
```

Input given: a=10, b=20

OUTPUT SCREENSHOT:

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

ALGORITHM:

1) Declare variables length, breadth and area in class Area. Take constructor Area(int l, int b).

2) Write a method calc() to calculate area=length*breadth.

3) Make class printclass friend of class X.

4) In printclass class in the function calc_area() show the value of area.

5) Display the output.

PROGRAM:

```
#include<iostream>

#include<string>

using namespace std;

class Area{

        int length,area,breadth;

        public:

                Area(int l,int b){

                        length=l;
```

```cpp
                    breadth=b;

            }

            void calc(){

                    area=length*breadth;

            }

            friend class printclass;


};
class printclass{

        public:

                void calc_area(Area obj){

                        cout<<"Area is= "<<obj.area;

                }

};


int main()

{

        Area obj(10,20);

        obj.calc();

        printclass p;

        p.calc_area(obj);


        return 0;

}
```
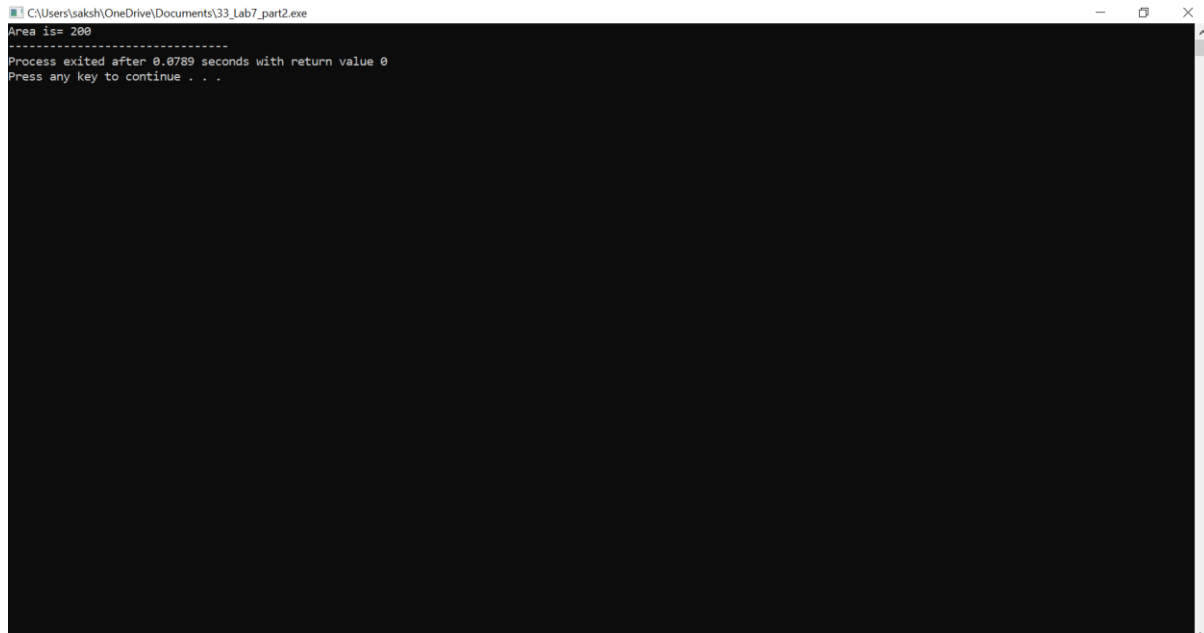
Input given: length =10 , breadth = 20

OUTPUT SCREENSHOT:

```
C:\Users\saksh\OneDrive\Documents\33_Lab7_part2.exe

Area is= 200
-------------------------------
Process exited after 0.0789 seconds with return value 0
Press any key to continue . . .
```