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:	**Sarika Laxmikant Galphade**
Roll No :	**36**

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.

There are 3 access specifiers for a class/struct/Union in C++. These access specifiers define how the members of the class can be accessed. Of course, any member of a class is accessible within that class(Inside any member function of that same class). Moving ahead to type of access specifiers, they are:
Public - The members declared as Public are accessible from outside the Class through an object of the class.
Protected - The members declared as Protected are accessible from outside the class BUT only in a class derived from it.
Private - These members are only accessible from within the class. No outside Access is allowed.

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

A friend function is a function that is specified outside a class but has the ability to access the class members' protected and private data. A friend can be a member's function, function template, or function, or a class or class template, in which case the entire class and all of its members are friends.

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

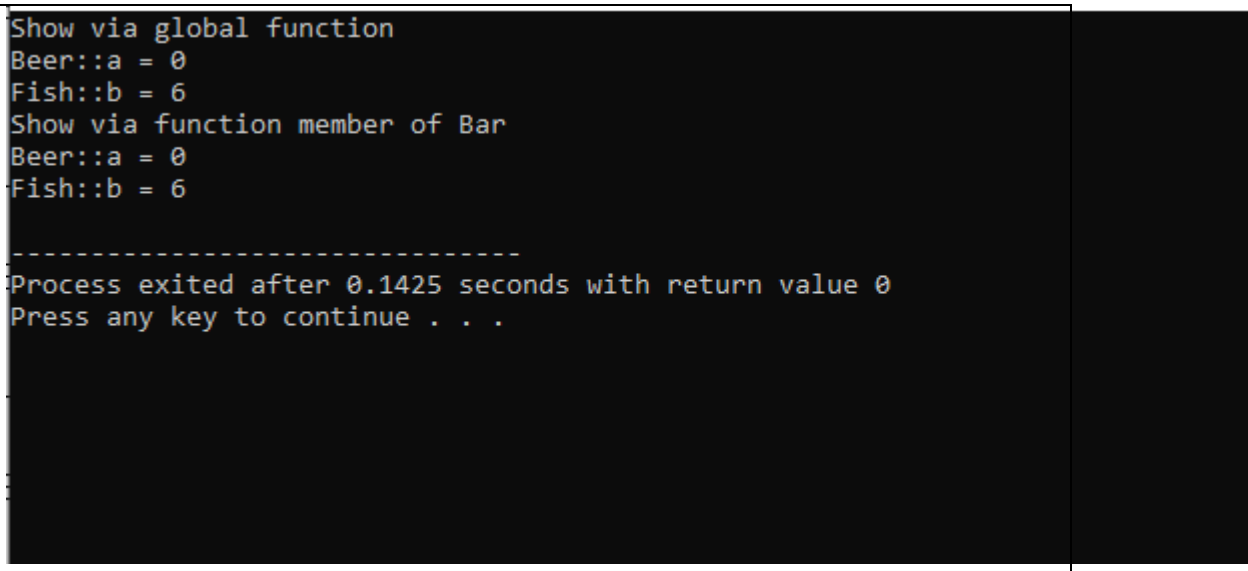
A friend class can have access to the data members and functions of another class in which it is declared as a friend. They are used in situations where we want a certain class to have access to another class's private and protected members.

Algorithm :	STEP 1: Start the program. STEP 2: Declare the class name as Base with data members and member functions. STEP 3: The function get() is used to read the 2 inputs from the user. STEP 4: Declare the friend function mean(base ob) inside the class. STEP 5: Outside the class to define the friend function and do the following. STEP 6: Return the mean value (ob.val1+ob.val2)/2 as a float. STEP 7: Stop the program.

Program :	#include <iostream>
	using namespace std;
	
	class Fish; // Forward declaration of class Foo in order for example to compile.
	class Beer {
	private:
	int a = 0;
	public:
	void show(Beer& x, Fish& y);
	friend void show(Beer& x, Fish& y); // declaration of global friend
	};
	
	class Fish {
	private:
	int b = 6;

```
  public:
     friend void show(Beer& x, Fish& y); // declaration of global friend
     friend void Beer::show(Beer& x, Fish& y); // declaration of friend from other class
};


// Definition of a member function of Bar; this member is a friend of Foo
void Beer::show(Beer& x, Fish& y) {
  cout << "Show via function member of Bar" << endl;
  cout << "Beer::a = " << x.a << endl;
  cout << "Fish::b = " << y.b << endl;
}


// Friend for Bar and Foo, definition of global function
void show(Beer& x, Fish& y) {
  cout << "Show via global function" << endl;
  cout << "Beer::a = " << x.a << endl;
  cout << "Fish::b = " << y.b << endl;
}


int main() {
   Beer a;
   Fish b;


   show(a,b);
   a.show(a,b);
}
```

Input given:	-

Output Screenshot:	

Algorithm :	STEP 1: Start the program. STEP 2: Declare the class name as Base with data members and member functions. STEP 3: The function get() is used to read the 2 inputs from the user. STEP 4: Declare the friend function mean(base ob) inside the class. STEP 5: Outside the class to define the friend function and do the following. STEP 6: Return the mean value (ob.val1+ob.val2)/2 as a float. STEP 7: Stop the program.
Program:	#include <iostream> using namespace std; class XYZ { private:

```
  char ch='Z';
  int num = 11;
public:

  friend class ABC;
};
class ABC {
public:
  void disp(XYZ obj){
    cout<<obj.ch<<endl;
    cout<<obj.num<<endl;
  }
};
int main() {
  ABC obj;
  XYZ obj2;
  obj.disp(obj2);
  return 0;
}
```

Input given: -

Output Screenshot: