**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:** | **Bhagyesh Subhash Manjrekar** |
| **Roll No :** | **07** |

## 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:

- Percept the Utility and applicability of OOP

**Faculty: Ms. Deepali Kayande**

## Theory:

## (1) Explain in details about access specifiers: public, private and protected:

**Ans:**

Access specifiers define how the members (attributes and methods) of a class can be accessed. For example, the class members are grouped into sections, private protected and public. These keywords are called access specifiers which define the accessibility or visibility level of class members.

By default, the class members are private, if the visibility labels are missing then by default all the class members are private.

In inheritance, it is important to know when a member function in the base class can be used by the objects of the derived class. This is called accessibility and the access specifiers are used to determine this.

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.

In this case, the members of the base class can be used only within the derived class and cannot be accessed through the object of derived class whereas they can be accessed by creating a function in the derived class.

**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.

In this case, the members of the base class can be used only within the derived class as protected members except for the private members.

## (2) Explain about friend function and friend classes in C++

**Ans:**

Non-member functions of a class will not have access to the private data of another class. There could be situations where we want two classes to share some functions and the data members. In that case, we can make the function a friend of these classes, and that will enable the function to access the private and protected data members of the classes.

**Friend Function:**

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. A friend function can be a member of another class or a global function.

**Special features of friend functions:**

- The friend function can be a member of another class or a function that is outside the scope of the class.
- A friend function can be declared in the private or public part of a class without changing its meaning.
- Friend functions can use objects of the class as arguments.
- A friend function cannot explicitly access member names directly. Every member name has to use the object's name and dot operator.

**Syntax:**

```
class className
{   // Other Declarations
  friend returnType functionName(arg list);
};
```

## Friend Class:

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.

Classes declared as friends to any another class will have all the member functions become friend functions to the friend class. Friend functions are used to work as a link between the classes.

## Syntax of Friend class:

```
class P
{
// Other Declarations
  friend class S;
};
class S
{
// Declarations
};
```

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

**Algorithm :**

**Step 1:** Start
**Step 2:** Create a class salary representing salary, getdata() function to print base salary.
**Step 3:** Create a constructor of class salary , add friend function in this class.
**Step 4:** Define friend function to access the private members from the friend function.
**Step 5 :** Create object for class salary, and print salary details of given employee.
**Step 6:** Stop

**Program:**

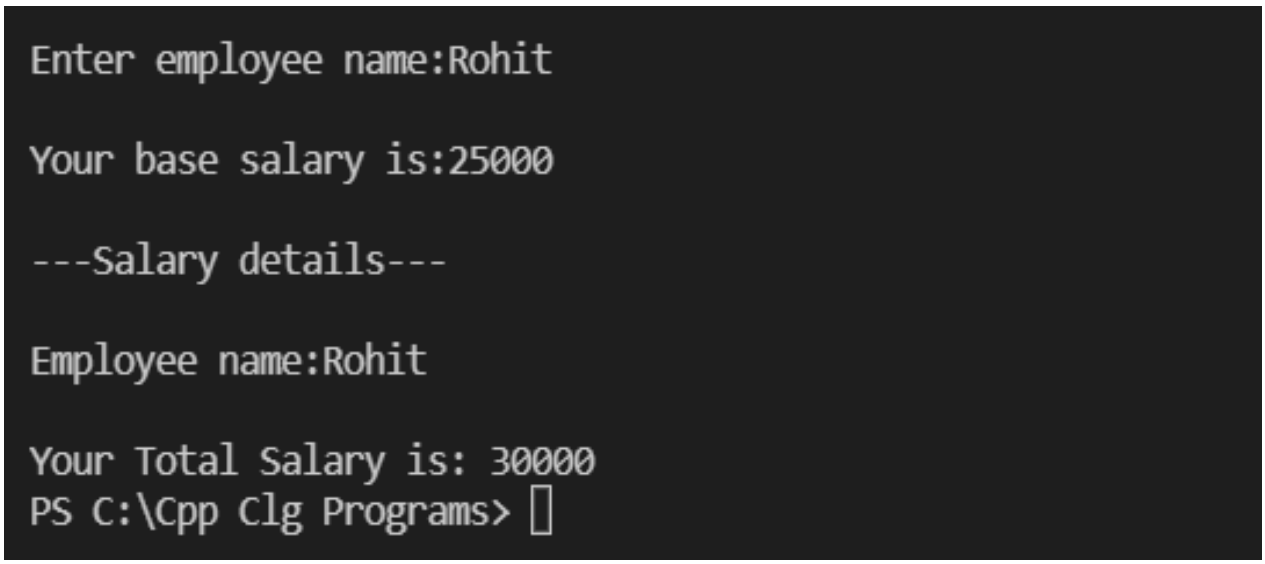
```
// C++ program to demonstrate the working of friend function

#include <iostream>
using namespace std;

class salary
{
private:
int sal;

public:
int getdata()
{
cout<<"\nYour base salary is:"<<sal<<endl;
return 0;
}
salary()
{
sal=25000;
}
// friend function
friend int addbonus(salary);
};

// friend function definition
```

```
int addbonus(salary s)
{
//accessing private members from the friend function
s.sal +=5000;
return s.sal;
}

int main()
{
salary s;
string name;
cout<<"\nEnter employee name:";
cin>>name;
s.getdata();
cout<<"\n---Salary details---"<<endl;
cout<<"\nEmployee name:"<<name<<endl;
cout<<"\n";
cout << "Your Total Salary is: "<<addbonus(s);

return 0;
}
```

**Input given:**

Enter employee name: Rohit
Your base salary is: 25000

**Output Screenshot:**



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

**Algorithm :**

**Step 1:** Start
**Step 2:** Create class X,Y representing value of digit1 and digit2.
**Step 3:** Declaring friend class declaration in class X, declaring constructors in class X,Y .
**Step 4:** Class Y being a friend class enables us to create objects of Class X inside of Class Y, then we will declare the multiply function in class Y to print the result.
**Step 5:** Stop

**Program:**

```
//C++ program to demonstrate the working of friend class
#include <iostream>
using namespace std;

// forward declaration
class ClassY;

class ClassX
{
int digit1;
public:
ClassX()
{
digit1=50;
}

friend class ClassY;  // friend class declaration
};

class ClassY
{
int digit2;
public:
int mul;
ClassY()
```

```
{
digit2=15;
}

int multiply()
{
ClassX m;
return m.digit1 * digit2;

}
};

int main()
{
ClassY n;
n.multiply();
cout<<"\nMultiplication: "<<n.multiply();
return 0;
}
```

**Input given:**

digit1=50
digit2=15

**Output Screenshot:**