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:	**6**
Student Name:	**Lovely Varshney**
Roll No :	**15**

Title:

1. To perform Multiple Inheritance in C++. Create a student class representing student roll number, name and branch and an exam class (derived class of student) representing the scores of the student in various subjects (maths, physics and chemistry) and sports class representing the score in sports. The sports and exam class isinherited by a result class which adds the exam marks and sports score to generate the final result.
2. To perform Hierarchical Inheritance in C++. Create an Employee class with attributes EmpID and EmpSalary. Also create necessary methods/constructors to accept these values from the user. Create classes permenantEmployee and TemporaryEmployee which will be derived classes of Employee. Mention hike attribute in these derived classes and calculate the total salary using generate_salary() method for respective types of employees. Objects of the derived classes should be created and salaries for the permanent and temporary employees should be calculated and displayed on the screen.

Learning Objective:

- Students will be able to perform multiple inheritance using C++.

Learning Outcome:

- Understanding the inheritance concept and reusability of the code.

Course Outcome:

ECL304.2	Comprehend building blocks of OOPs language, inheritance, package and interfaces

Theory:

- Explain in details about inheritance, its types, syntaxes and block diagrams.

Algorithm 1 :	**1 – Create a parent class student and initialize its data members.** **2- Create the derived class of student class - exam class to take input of marks** **3 - Create exam class to take input of marks** **3 – Create the result class to inherit the exam class and sports class**

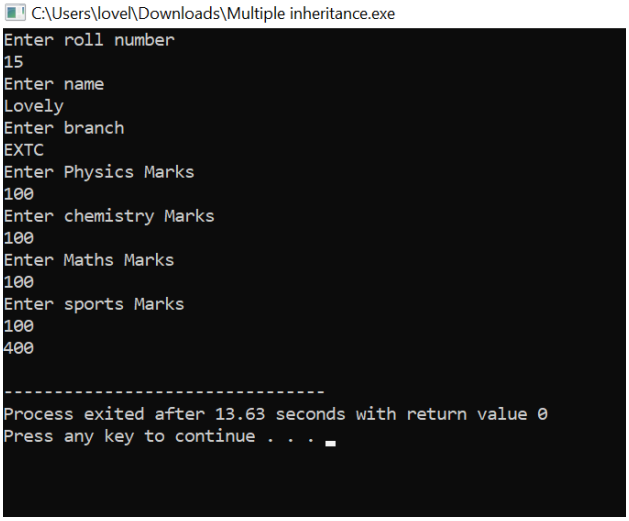
	publically and to calculate the total. **4 – Create the main function to call the class functionality and display the result.s**
Program 2 :	**#include<iostream>** **using namespace std;** **class student{** **public:** **int roll_number;** **string name;** **string branch;** **student(){** **cout<<"Enter roll number"<<endl;** **cin>>roll_number;** **cout<<"Enter name"<<endl;** **cin>>name;** **cout<<"Enter branch"<<endl;** **cin>>branch;** **}** **};** **class exam: public student{** **public:** **int maths;** **int physics;** **int chemistry;** **exam(){** **cout<<"Enter Physics Marks"<<endl;** **cin>>physics;**

```
        cout<<"Enter chemistry Marks"<<endl;
        cin>>chemistry;
        cout<<"Enter Maths Marks"<<endl;
        cin>>maths;
  }
};

class sport{
        public:
        int sports;
        sport(){
        cout<<"Enter sports Marks"<<endl;
        cin>>sports;
  }
};

class result : public exam, public sport {
        public:
                int total;
                result(){
        total = maths + physics + chemistry + sports;
                cout<<total<<endl;
           }
};

int main()
{
        result obj;
```

	return 0; **}**
Input given 1:	**ROLL NO – 15** **BRANCH -EXTC** **PHYSICS MARKS -100** **CHEMISTRY MARKS -100** **MATHS MARKS – 100** **SPORTS MARKS – 100** **TOTAL = 400**
Output Screenshot 1:	
Algorithm 2:	1. **Creating the parent class employee and initialize its data members.(EmpId ,EmpSalary) and a basic function get details() to print the details.** 2. **Create 2 child class permanent employee and temporary employee that inherit employee class publically.** 3. **In this classes , create generate salary() that return the employee salary + hike in their salary** 4. **In main function, Create the object of derived class and print their respective details.**
Program 2:	**#include <iostream>** **using namespace std;**

```
class Employee
{
        protected:
        string EmpID = "";

        double Empsalary;

        public:
                Employee()
                {
        cout<<endl<<"Enter the employee id : ";
        cin>>EmpID;
                        cout<<endl<<"Enter the employee Sal : ";
        cin>>Empsalary;

                }

        void getDetails()
        {
                cout <<endl<< "EmployeeID is : " << EmpID;
                cout <<endl<<endl<<"Employee Total Salary is : " <<
Empsalary;
        }


};
```

```
class Permanent_Employee: public Employee
{
        double hike;

        public:

        Permanent_Employee( double increment)
        {
                hike = increment;
        }

   void getDetails()
        {
                cout <<endl<< "EmployeeID is : " << EmpID;
                cout <<endl<<endl<<"Employee Total Salary is : " <<
generate_salary();
        }

        float generate_salary()
        {
                return (Empsalary + hike);
        }
};

class Temporary_Employee: public Employee
{
        double hike;
```

```
        public:


        Temporary_Employee( double increment)
        {
                hike = increment;
        }


   void getDetails()
        {
                cout <<endl<< "EmployeeID is : " << EmpID;
                cout <<endl<<endl<<"Employee Total Salary is : " <<
generate_salary();
        }


        float generate_salary()
        {
                return (Empsalary + hike);
        }
};


int main()
{


   cout<<endl<<"For Temporary Employee: "<<endl;
        Temporary_Employee T(667.6);
   cout << endl << endl << "Details of Temporary_Employee : " <<
endl;
        T.getDetails();
   cout<<endl<<endl<<"For Permanent Employee: "<<endl;
```

	Permanent_Employee P(777.99); **cout << endl << endl << "Details of Permanent_Employee : " << endl;** **P.getDetails();** **return 0;** **}**
Input given 2:	**For temporary employee –** **Employee id – 212** **Employee sal – 2000** **For permanent employee –** **Employee id – 312** **Employee sal – 5000**
Output Screenshot 2: