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:	**Rakshita Rajeev Khnatwal**
Roll No :	**26**

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

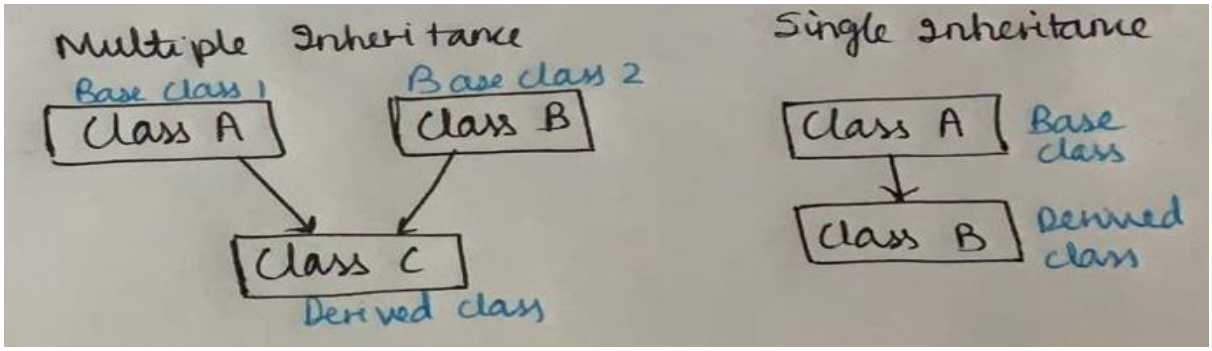
Theory:

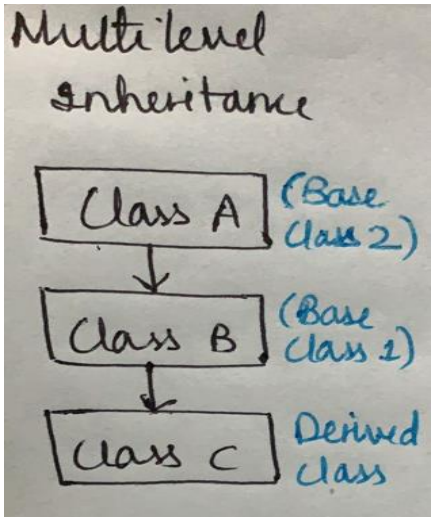
- Explain in details about inheritance, its types, syntaxes and block diagrams.

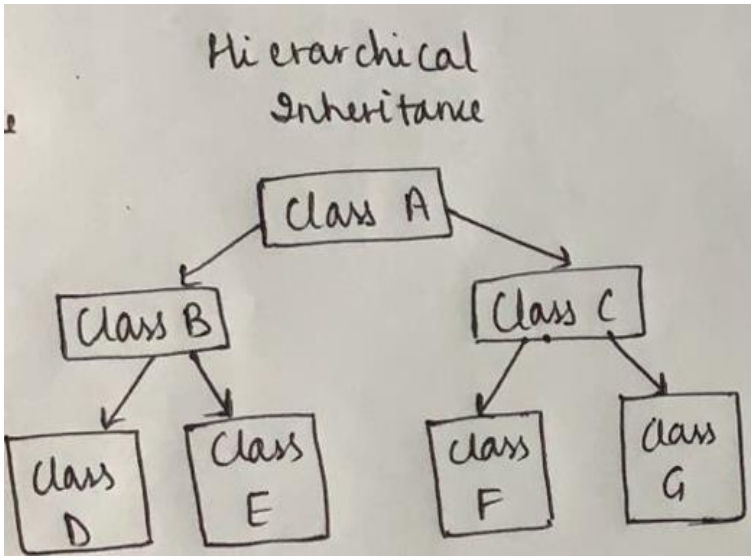
Inheritance is one of the most important feature of Object Oriented Programming.
Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.
Super Class: The class whose properties are inherited by sub class is called Base Class or Super class.

1. Single Inheritance: In single inheritance, a class is allowed to inherit from only one class. i.e. one sub class is inherited by one base class only.
2. Multiple Inheritance: Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. i.e one sub class is inherited from more than one base classes.

3.Multilevel Inheritance: In this type of inheritance, a derived class is created from another derived class.
4. Hierarchical Inheritance: In this type of inheritance, more than one sub class is inherited from a single base class. i.e. more than one derived class is created from a single base class.







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

1.

Algorithm :	STEP 1: Start STEP 2: Make a student class STEP 3: Define attributes roll no, name, branch. Declaring method getdata STEP 4: Make a childclass Exam of class student STEP 5: Define attributes math, physics, chemistry. Declaring method getmarks STEP 6: Make a class sports STEP 7: Declaring attribute sportscore and method getsportsmarks STEP 8: Make subclass Result inherited from class student and exam STEP 9:Creating objects of class in main function STEP 10: Calling methods using object of class STEP 11: Display result STEP 12: Stop
Program:	/* 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. */ #include<iostream> #include<conio.h> using namespace std;

```
class student {
   protected:
      int rollno;
      char name[50];
      char branch[10];

   public:
      void getdata()
      {
         cout << "Enter your roll no.";
         cin >>rollno;
         cout <<"Enter your name:";
         cin>>name;
         cout<<"Enter your branch:";
         cin>>branch;
      }
};

class exam : public student {
   public:
      int math, physics, chemistry;
   public:
      void getmarks() {
         cout<<"Enter marks in maths:";
         cin>>math;
         cout<<"Enter marks in physics:";
         cin>>physics;
         cout<<"Enter marks in chemistry:";
```

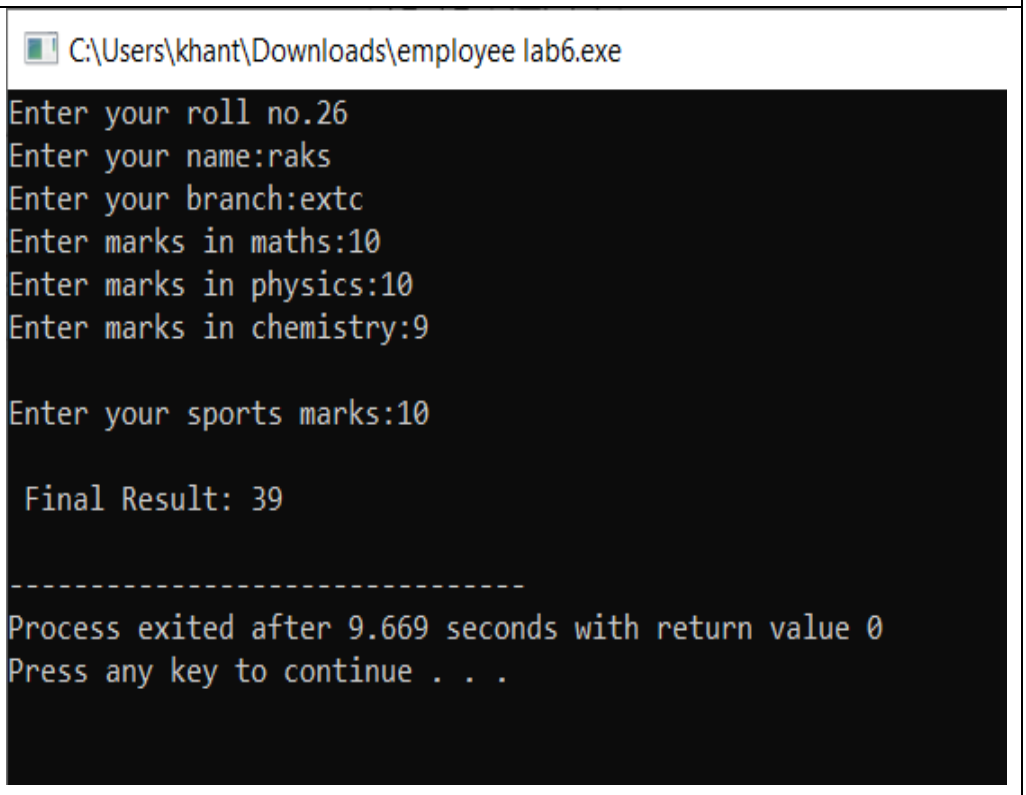
```
        cin>>chemistry;
    }
};

class sports {
  public:
    int sportscore;
  public:
    void getsportsmarks() {
        cout<<endl<<"Enter your sports marks:";
        cin>>sportscore;
    }
};

class result: public exam, public sports{
  float finalresult;
  public:
  void display(){
    finalresult = math+physics+chemistry+sportscore;
    cout<<"\n Final Result: "<< finalresult<<endl;
  }
};

int main(){
  student s1;
  s1.getdata();
  result obj;
  obj.getmarks();
```

	obj.getsportsmarks();
	obj.display();
	
	return 0;
	}

Input given:	Roll no: 26 Name: Raks Branch: EXTC Marks in math: 10 Marks in Physics: 10 Marks in Chemistry: 9 Marks in Sports: 10
Output Screenshot:	

2.

Algorithm :	STEP 1: Start STEP 2: Create an employee class STEP 3:Define attributes EmpID, ImpSalary, declaring method set_details(). STEP 4: Derive classes Temperory employee STEP 5:Define attributes, declare method calculate_details(), print_details(). STEP 6: Derive class Permanent Employee STEP 7:Define attributes, declare method calculate_details(), print_details(). STEP 8:Creating objects of class in main function STEP 9:Calling methods using object of class STEP 10:Display salary of employee STEP 11: Stop
Program:	**/* 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. */ #include<iostream> using namespace std; class Employee{**

```
public:
  int EmpID;
  char Employee_name[50];
  float EmpSalary;
  float DA;
  float IT;
  float Salary;

public:
  void set_details()
  {
    cout<<"Enter Employee ID"<<endl;
    cin>> EmpID ;
    cout<<"Enter Employee name "<<endl;
    cin>>Employee_name;
  }
public:
  int get_empid() {
    return this->EmpID;
  }
};



class PermanentEmployee : public Employee{
public:
  void calculate_details()
  //basic salary for permanent employee= 30000
  {
```

```
   DA=1.32*30000;
   IT=0.30*(30000+DA);
   Salary=(30000+DA)-IT;
  }
public:
  void print_details()
  {
   cout<<"Employ Basic Salary : "<<30000<<endl;
   cout<<"Employ DA : "<<DA<<endl;
   cout<<"Employ IT : "<<IT<<endl;
   cout<<"Employ Salary : "<<Salary<<endl;
  }
};
class TemperoryEmployee : public Employee{
  //basic salary for temperory employee= 20000;
public:
  void calculate_details()
  {
   DA=1.32*20000;
   IT=0.30*(20000+DA);
   Salary=(20000+DA)-IT;
  }

  void print_details()
  {
   cout<<"Employ Basic Salary : "<<20000<<endl;
   cout<<"Employ DA : "<<DA<<endl;
   cout<<"Employ IT : "<<IT<<endl;
```

	cout<<"Employ Salary : "<<Salary<<endl; **}** **};** **int main(){** **Employee e;** **e.set_details();** **int a = e.get_empid();** **if(a<=10) {** **PermanentEmployee pe;** **pe.calculate_details();** **pe.print_details();** **cout<<"permanent employee"<<endl;** **}** **else {** **TemperoryEmployee te;** **te.calculate_details();** **te.print_details();** **cout<<"temp employee"<<endl;** **}** **return 0;** **}**
Input given:	Employee ID: 12 Employee name:Raks

Output Screenshot :