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:	**Simran Santosh Koparkar**
Roll No :	**41**

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:

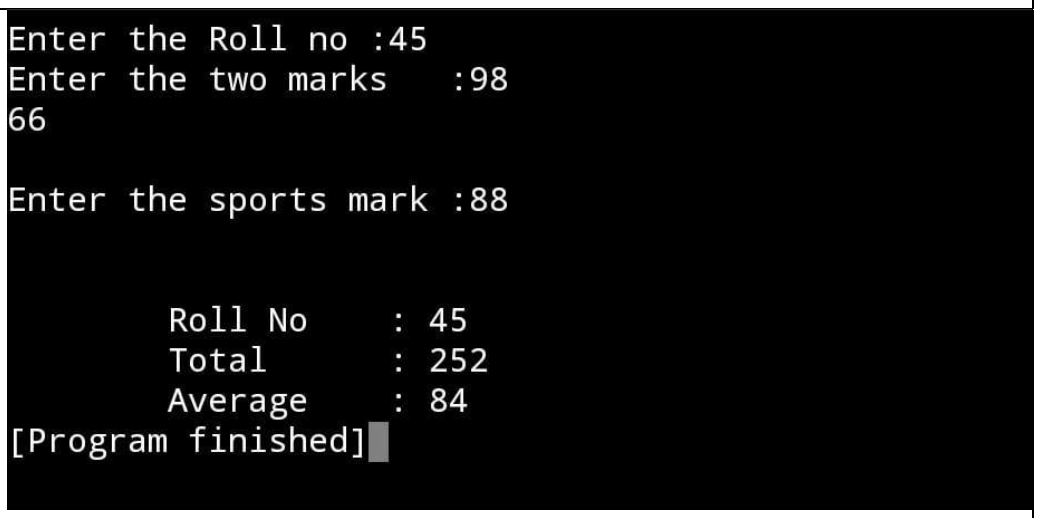
- Explain in details about inheritance, its types, syntaxes and block diagrams.

Inheritance is a mechanism of acquiring the features and behaviors of a class by another class. The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class. Single Inheritance. Multilevel Inheritance. Hierarchical Inheritance. Multiple Inheritance

Faculty: Ms. Deepali Kayande

Algorithm :	**Step 1: Start the program.** **Step 2: Declare the base class student.** **Step 3: Declare and define the function get() to get the student details.** **Step 4: Declare the other class sports.** **Step 5: Declare and define the function getsm() to read the sports mark.** **Step 6: Create the class statement derived from student and sports.** **Step 7: Declare and define the function display() to find out the total and average.** **Step 8: Declare the derived class object,call the functions get(),getsm() and display().** **Step 9: Stop the program.**
Program:	**#include<iostream.h>** **#include<conio.h>** **class student {** **protected:** **int rno, m1, m2;** **public:** **void get() {** **cout << "Enter the Roll no :";** **cin>>rno;** **cout << "Enter the two marks :";** **cin >> m1>>m2;** **}** **};** **class sports {** **protected:**

```
   int sm; // sm = Sports mark
public:

   void getsm() {
      cout << "\nEnter the sports mark :";
      cin>>sm;

   }
};

class statement : public student, public sports {
   int tot, avg;
public:

   void display() {
      tot = (m1 + m2 + sm);
      avg = tot / 3;
      cout << "\n\n\tRoll No    : " << rno << "\n\tTotal     : " << tot;
      cout << "\n\tAverage    : " << avg;
   }
};

void main() {
   clrscr();
   statement obj;
   obj.get();
   obj.getsm();
   obj.display();
```

	getch(); **}**
Input given:	**45** **98** **66** **88**
Output Screenshot:	

Program:	**#include<iostream.h>** **#include<fstream.h>** **#include<conio.h>** **#include<stdio.h>** **//class Employee** **class Employee** **{** **public:** **char name[30];** **int salary;**

```
};


//class Manager inherited from class Employee
class Manager:public Employee
{
public:
char dept[30];


//to getdata from the user
void getData()
{
cout<<"\nEnter the following information about manager:\nName :";
gets(name);
cout<<"\nDepartment :";
gets(dept);
cout<<"\nSalary :";
cin>>salary;
}


//to show data to the user
void show()
{
cout<<"\n Manager Name :"<<name<<"\nDepartment
:"<<dept<<"\nSalary :"<<salary;
}
};


//class Executive inheriting Manager class
```

```
class Executive:public Manager
{
public:
Executive()
{
cout<<"\nExecutive";
getData();
show();
}
};

//main() function for testing
void main()
{
clrscr();
Executive e;
getch();
}
[15:49, 11/5/2021] Simran K: #include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>



class Employee
{
public:
char name[30];
```

```
int salary;
};

//class Manager inherited from class Employee
class Manager:public Employee
{
public:
char dept[30];

void getData()
{
cout<<"\nEmployee Name :";
gets(name);
cout<<"\nDepartment :";
gets(dept);
cout<<"\nSalary :";
cin>>salary;
}

//to show data to the user
void show()
{
cout<<"\nEmployee Name :"<<name<<"\nDepartment
:"<<dept<<"\nSalary :"<<salary;
}
};
```

	class Executive:public Manager **{** **public:** **Executive()** **{** **cout<<"\nPermanent Employee";** **getData();** **show();** **}** **};** **void main()** **{** **clrscr();** **Executive e;** **getch();** **}**
Input given:	**Nithin** **EXTC** **50000**

Output Screenshot: