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:	Umer rafique shaikh
Roll No :	44

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 is inherited by a result class which adds the exam marks and sports score to generate the final result.

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 detail about inheritance, its types, syntaxes and block diagrams.

1) The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as base class and the new class is referred to as the derived class. C++ strongly supports the concept of reusability. This is basically done by creating new classes, reusing the properties of the existing ones. Functions and variables of a class that has been tested can be used by object of another class. This is known as inheritance.The reuse of a class that has already been tested; debugged and used many times can save the efforts of developing and testing the same again

One of the most useful features of classes is inheritance. It is possible to declare a class that inherits the properties of another

class or classes. This means that with good class design, you can build applications, which are based on proven re-usable code

le main purpose behind inheritance is code reusability

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 students 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> 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:	44 umer extc 10 12 13 14

output

```
/tmp/BC43WnFUWZ.o
Enter roll number
44
Enter name
umer
Enter branch
extc
Enter Physics Marks
10
Enter chemistry Marks
12
Enter Maths Marks
13
Enter sports Marks
14
49
```