

**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**

<b>Lab Number:</b>	<b>04</b>
<b>Student Name:</b>	SHIVAM SHARMA
<b>Roll No :</b>	E-19

**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.
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 permanentEmployee 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++.
- Students will be able to perform hierarchical inheritance using C++.

**Learning Outcome:**

- Understanding the inheritance concept and reusability of the code.

**Theory:**

What is Inheritance?

**Faculty: Ms. Deepali Kayande**

**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**

Inheritance in C++ is a vital concept and can not be overlooked. Understanding inheritance is critical for understanding the whole point behind object-oriented programming. For instance, you are a human. You inherit from the class 'Humans' characteristic features, such as walking, sitting, running, eating, and so on. The class 'Humans' inherits these characteristic features from the class 'Mammal' which makes the 'Human' class a derived class of 'Mammal'. This 'Mammal' class inherits its characteristic features from another class 'Animal' which makes the 'Mammal' class a derived class of the class 'Animal' and makes the 'Animal' a base class.

One of the most astonishing features of inheritance is code reusability. This reusability also provides you with clean code, and the replication of code gets reduced to almost zero.

Reusing existing codes serves various advantages. It saves time, money, effort, and increases a program's reliability.

#### **Uses of Inheritance:**

Inheritance is a useful concept of object-oriented programming. Inheritance in C++ serves many advantages. There are several reasons why inheritance was introduced in OOPs. You will see some of the major reasons behind the introduction of inheritance in C++, below:

- Another reason is the idea of reusability. Code reusability ensures that a clean code is provided to the programmer. This also helps in the reduction of rewriting and serves as a bug-free code, as the replication of the code gets reduced to almost zero with the help of reusability. Other advantages of reusability are time management, maintenance, and ease of extension. You can do manipulations and add some desired features to a class that already exists through inheritance.
- Inheritance increases the relatability of the code to real-world scenarios drastically.
- One more reason is the transitive nature of inheritance. Transitive nature implies that if two objects that are in succession show a pattern, then all the objects of that order must show the exact pattern. For example, if a new class TataSafari is declared as a subclass of Car, which itself is a subclass of Vehicle, then TataSafari must also be a Vehicle i.e., inheritance is transitive in nature.

#### **Types Of Inheritance:**

C++ supports five types of inheritance:

1. Single inheritance

**Faculty: Ms. Deepali Kayande**

**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**

2. Multiple inheritance
3. Hierarchical inheritance
4. Multilevel inheritance
5. Hybrid inheritance

**Algorithm 1:**

Step 1 : start

Step 2 : Create a parent class student and initialize its data members.

Step 3 : Create the derived class of student class - exam class to take input of marks

Step 4 : Create sports class to take input of marks

Step 5 : Create the result class to inherit the exam class and sports class publically and to calculate the total.

Step 6 : Create the main function to call the class functionality and display the results

Step 7 : end

**Program 1:**

**Input:**

```
#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.";
```

**Faculty: Ms. Deepali Kayande**

**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**

```
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;
}
};
```

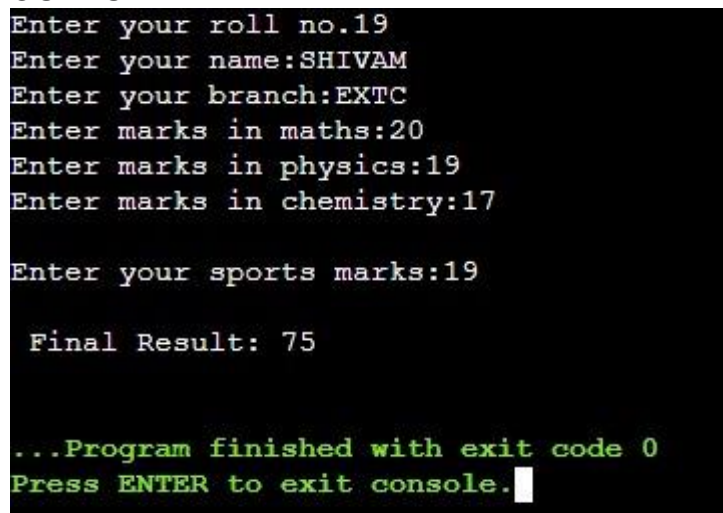
```
class result: public exam, public sports{
float finalresult;
public:
void display(){
finalresult = math+physics+chemistry+sportscore;
cout<<"\n Final Result: "<< finalresult<<endl;
}
};
```

```
int main(){
student s1;
```

**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**

```
s1.getdata();  
result obj;  
obj.getmarks();  
obj.getsportsmarks();  
obj.display();  
  
return 0;  
}
```

## OUTPUT



```
Enter your roll no.19  
Enter your name:SHIVAM  
Enter your branch:EXTC  
Enter marks in maths:20  
Enter marks in physics:19  
Enter marks in chemistry:17  
  
Enter your sports marks:19  
  
Final Result: 75  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

## PROGRAM 2

### ALGORITHM 2

Step1 : start

Step 2: Creating the parent class employee and initialize its data members.(EmpId ,Emp Salary) and a basic function get details() to print the details.

Step 3 :Create 2 child class permanent employee and temporary employee that inherit employee class publically.

Step 4 : In this classes , create generate salary() that return the employee salary + hike in their salary

Step 5 : In main function, Create the object of derived class and print their respective details.

Step 6 : end

**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**

**INPUT :**

```
#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;
```

**Faculty: Ms. Deepali Kayande**

**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**

```
    cout<<"Employ IT : "<<IT<<endl;
    cout<<"Employ Salary : "<<Salary<<endl;
}
};
class TemporaryEmployee : public Employee{
    //basic salary for temporary 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 {
        TemporaryEmployee te;
        te.calculate_details();
        te.print_details();
        cout<<"temp employee"<<endl;
    }
    return 0;
}
```

**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**

**Output:**

```
Enter Employee ID
19
Enter Employee name
SHIVAM
Employ Basic Salary : 20000
Employ DA : 26400
Employ IT : 13920
Employ Salary : 32480
temp employee
```