

**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>6</b>
<b>Student Name:</b>	<b>Jaipreet Singh Saini</b>
<b>Roll No :</b>	<b>11</b>

**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++.

**Learning Outcome:**

- Understanding the inheritance concept and reusability of the code.

**Course Outcome:**

<b>ECL304.2</b>	Comprehend building blocks of OOPs language, inheritance, package and interfaces
-----------------	--

**Theory:**

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

Solution:- 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. Inheritance implements the **IS-A** relationship.

For example, mammal **IS-A** animal, dog **IS-A** mammal; Hence dog **IS-A** animal as well.

**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**

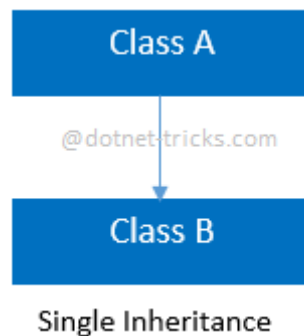
OOPs support the six different types of inheritance as given below :

1. Single inheritance
2. Multi-level inheritance
3. Multiple inheritance
4. Multipath inheritance
5. Hierarchical Inheritance
6. Hybrid Inheritance

### 1. Single inheritance

In this inheritance, a derived class is created from a single base class.

In the given example, Class A is the parent class and Class B is the child class since Class B inherits the features and behavior of the parent class A.



### Syntax for Single Inheritance

```
//Base Class  
  
class A  
{  
    public void fooA()  
    {  
        //TO DO:  
    }  
}
```

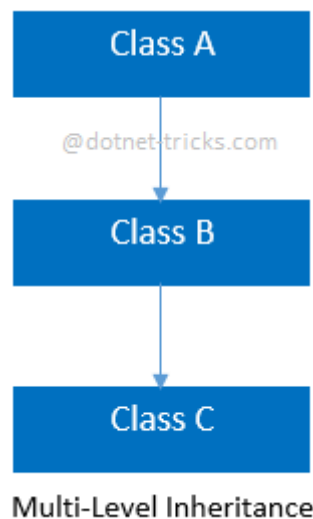
**Don Bosco Institute of Technology, Kurla(W)**  
**Department of Electronics and Tele-Communication Engineering**  
**ECL304 - Skill Lab: C++ and Java Programming**  
**Sem III**  
**2021-22**

```
//Derived Class  
  
class B : A  
{  
    public void fooB()  
    {  
        //TO DO:  
    }  
}
```

## 2. Multi-level inheritance

In this inheritance, a derived class is created from another derived class.

In the given example, class c inherits the properties and behavior of class B and class B inherits the properties and behavior of class A. So, here A is the parent class of B and class B is the parent class of C. So, here class C implicitly inherits the properties and behavior of class A along with Class B i.e there is a multilevel of inheritance.



### Syntax for Multi-level Inheritance

```
//Base Class  
  
class A  
{
```

**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**

```
public void fooA()

{

//TO DO:

}

}

//Derived Class

class B : A

{

public void fooB()

{

//TO DO:

}

}

//Derived Class

class C : B

{

public void fooC()

{

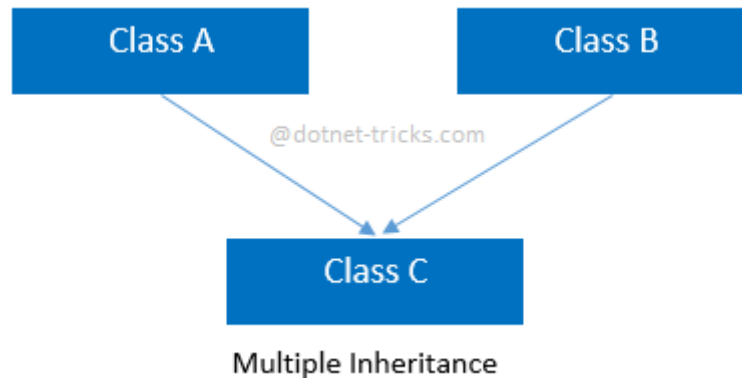
//TO DO:

}

}
```

### 3. Multiple inheritance

In this inheritance, a derived class is created from more than one base class. This inheritance is not supported by [.NET](#) Languages like C#, F# etc. and Java Language. In the given example, class c inherits the properties and behavior of class B and class A at same level. So, here A and Class B both are the parent classes for Class C.



#### Syntax for Multiple Inheritance

```
//Base Class
class A
{
    public void fooA()
    {
        //TO DO:
    }
}

//Base Class
class B
{
    public void fooB()
    {
```

**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**

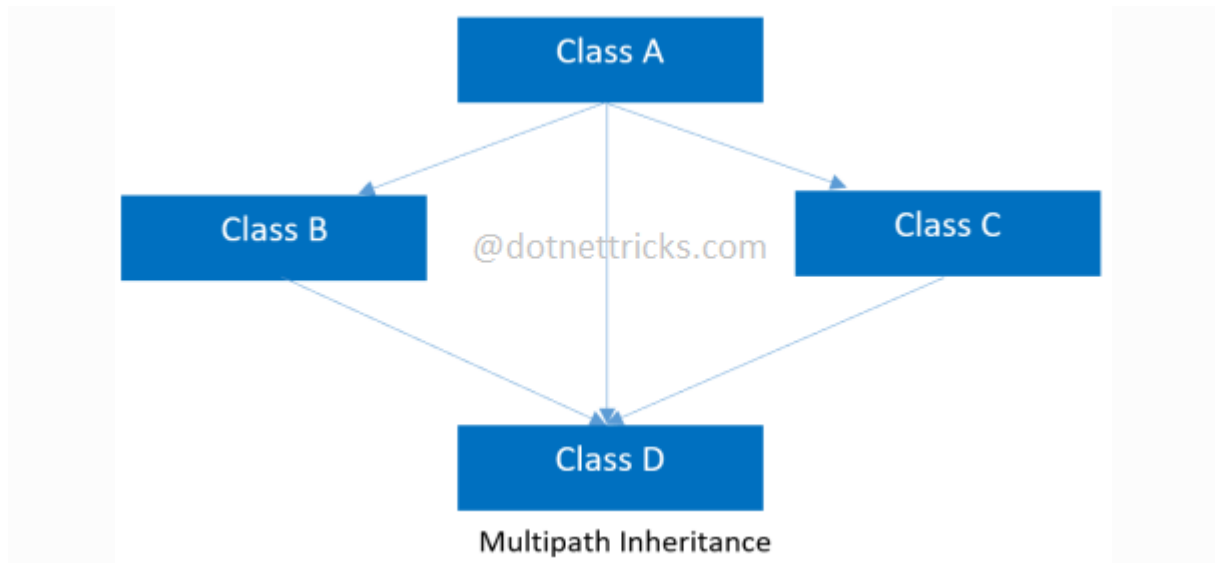
```
//TO DO:  
  
}  
  
}  
  
//Derived Class  
  
class C : A, B  
{  
    public void fooC()  
    {  
        //TO DO:  
    }  
}
```

#### 4. Multipath inheritance

In this inheritance, a derived class is created from another derived classes and the same base class of another derived classes. This inheritance is not supported by .NET Languages like C#, F# etc.

In the given example, class D inherits the properties and behavior of class C and class B as well as Class A. Both class C and class B inherits the Class A. So, Class A is the parent for Class B and Class C as well as Class D. So it's making it Multipath inheritance.

**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**



### Syntax for Multipath Inheritance

```
//Base Class  
  
class A  
{  
    public void fooA()  
  
    {  
        //TO DO:  
    }  
}  
  
//Derived Class  
  
class B : A  
{  
    public void fooB()  
  
    {  
        //TO DO:
```

**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**

```
}  
  
}  
  
//Derived Class  
  
class C : A  
{  
    public void fooC()  
    {  
        //TO DO:  
    }  
}  
  
//Derived Class  
  
class D : B, A, C  
{  
    public void fooD()  
    {  
        //TO DO:  
    }  
}
```

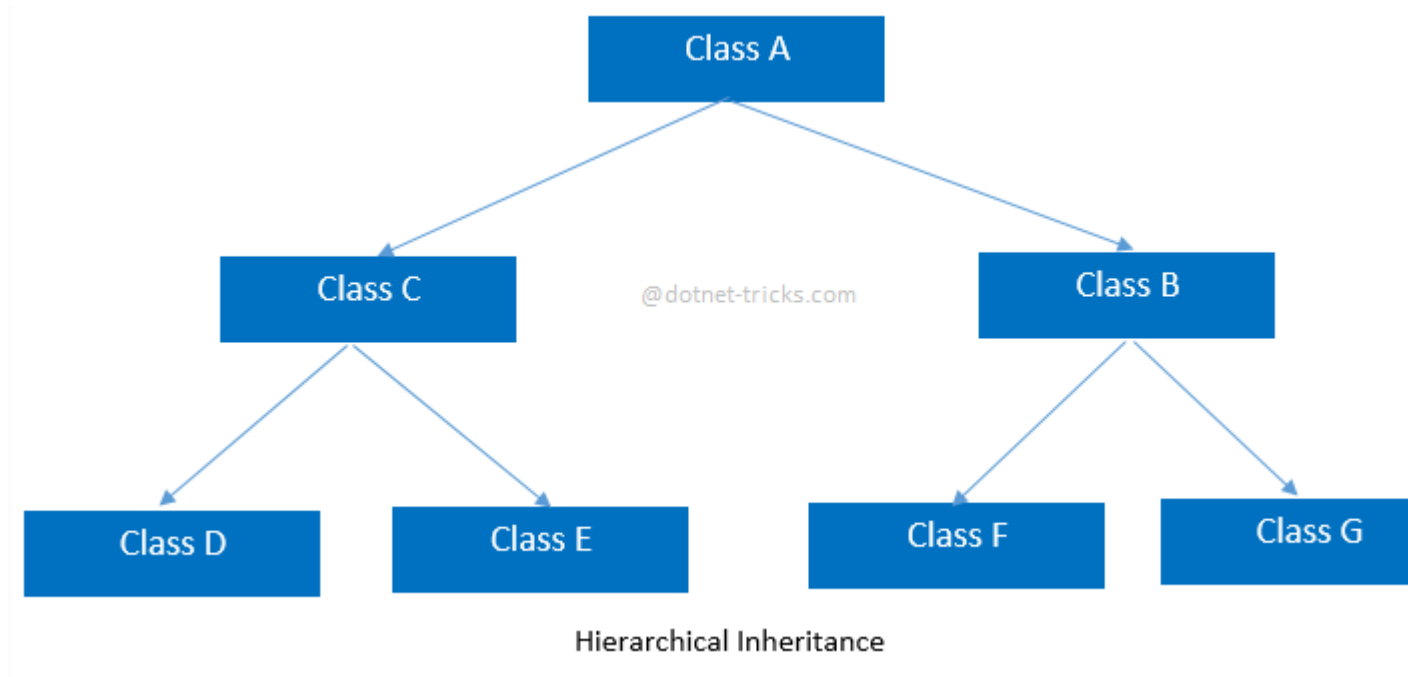
## 5. Hierarchical Inheritance

In this inheritance, more than one derived classes are created from a single base class and further child classes act as parent classes for more than one child classes.

In the given example, class A has two child classes B and class D. Further, class B and class C both are having two child classes - class D and E; class F and G respectively.



**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**



### Syntax for Hierarchical Inheritance

```
//Base Class  
  
class A  
{  
    public void fooA()  
    {  
        //TO DO:  
    }  
}  
  
//Derived Class  
  
class B : A  
{  
    public void fooB()
```

**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**

```
{  
  
//TO DO:  
  
}  
  
}  
  
  
//Derived Class  
  
class C : A  
  
{  
  
    public void fooC()  
  
    {  
  
        //TO DO:  
  
    }  
  
}  
  
  
//Derived Class  
  
class D : C  
  
{  
  
    public void fooD()  
  
    {  
  
        //TO DO:  
  
    }  
  
}  
  
  
//Derived 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**

```
class E : C
{
    public void fooE()
    {
        //TO DO:
    }
}
```

//Derived Class

```
class F : B
{
    public void fooF()
    {
        //TO DO:
    }
}
```

//Derived Class

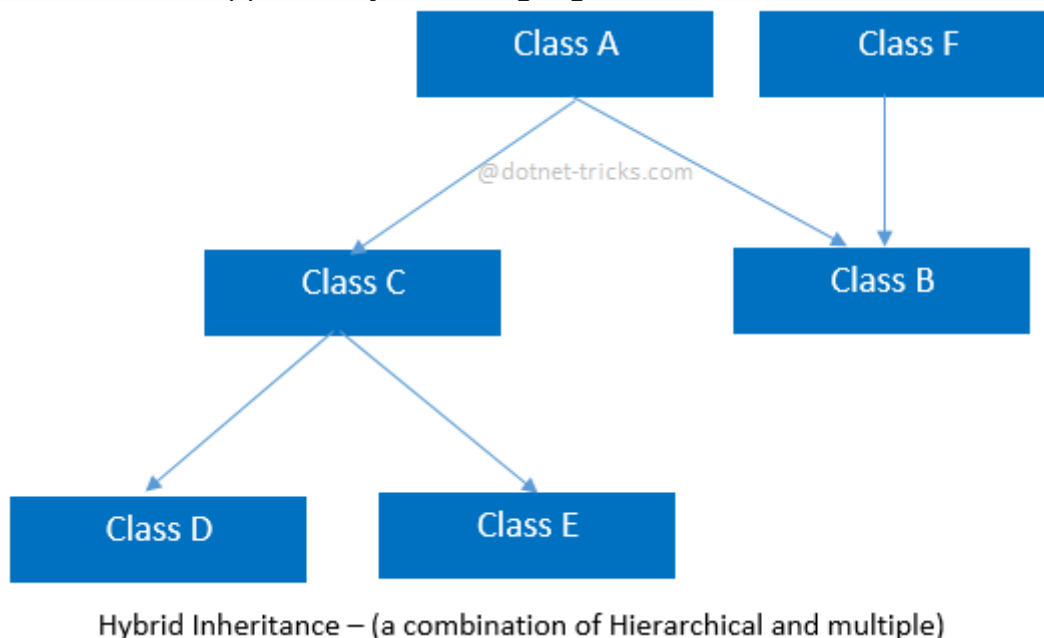
```
class G :B
{
    public void fooG()
    {
        //TO DO:
    }
}
```

}

## 6. Hybrid inheritance

This is combination of more than one inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical and Multipath inheritance or Hierarchical, Multilevel and Multiple inheritance.

Since .NET Languages like [C#](#), F# etc. does not support multiple and multipath inheritance. Hence hybrid inheritance with a combination of multiple or multipath inheritances is not supported by .NET Languages.



## Syntax for Hybrid Inheritance

```
//Base Class  
  
class A  
{  
    public void fooA()  
    {  
        //TO DO:  
    }  
}
```

**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**

```
//Base Class  
  
class F  
{  
    public void fooF()  
    {  
        //TO DO:  
    }  
}
```

```
//Derived Class  
  
class B : A, F  
{  
    public void fooB()  
    {  
        //TO DO:  
    }  
}
```

```
//Derived Class  
  
class C : A  
{  
    public void fooC()  
    {
```

**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**

```
//TO DO:

}

}

//Derived Class

class D : C

{

    public void fooD()

    {

        //TO DO:

    }

}

//Derived Class

class E : C

{

    public void fooE()

    {

        //TO DO:

    }

}
```

**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**

**6.0(Multiple) :- Input given a=11 b= 56 67**

**OUTPUT:**

```
C:\Users\jaipr\OneDrive\Desktop\Untitled2.exe
Enter the Roll no :11
Enter the two marks :56 67
Enter the sports mark :78

Roll No : 11
Total : 201
Average : 67
```

**6.1(Hierarchical) : - Input given:- a= ABC b= 123 c= M**

**OUTPUT:**

```
C:\Users\jaipr\OneDrive\Desktop\hierarchical.exe
Enter employee's basic info:
Enter Name: ABC
Enter Emp. Id: 123
Enter Gender: M
Enter employee's department info:
Enter Department Name: Testing
Enter assigned work: Test Game
Enter time in hours to complete work: 90
Employee's Information is:
Basic Information...:
Name: ABC
Employee ID: 123
Gender: M

Department Information...:
Department Name: Testing
Assigned Work: Test Game
Time to complete work: 90

-----
Process exited after 28.44 seconds with return value 0
Press any key to continue . . .
```