CSE-284: Object Oriented Programming

Inheritance in C++

Reference:
TrytoProgram.com

Eftekhar Hossain
Lecturer
Dept. of ETE, CUET

CUET

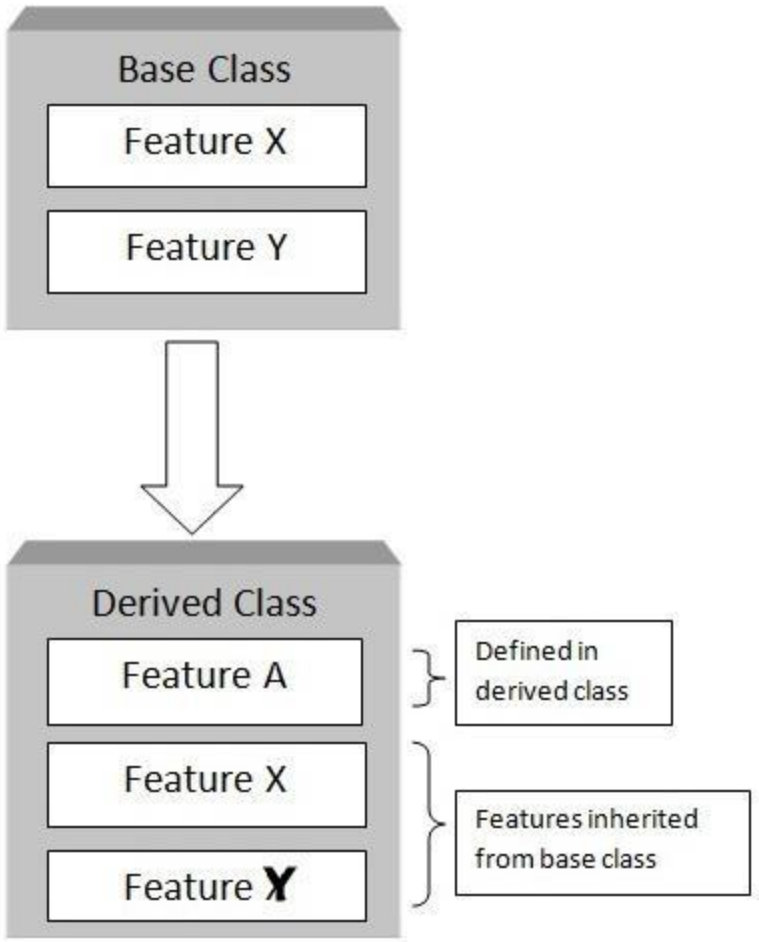
Topics to be Covered

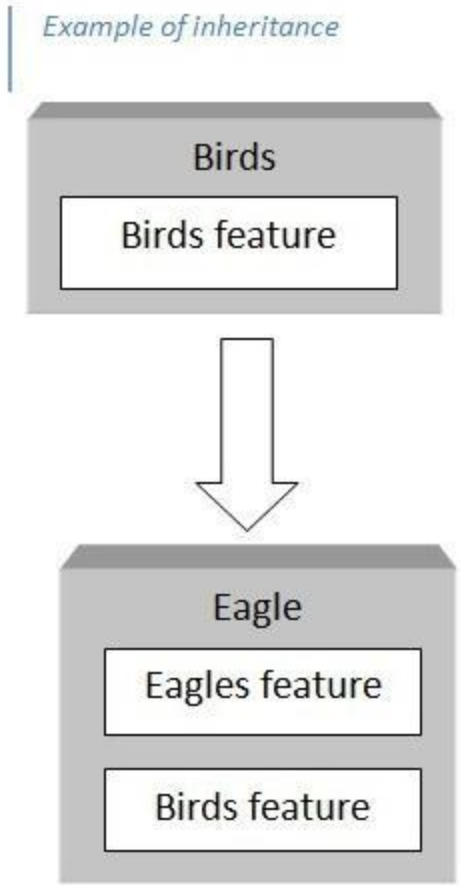
- What is Inheritance?
- Types of Inheritance
- What are the Access specifiers?

Inheritance

- C++ Inheritance is one of the powerful features of C++.
- Inheritance is the technique of building new classes called derived classes from the existing class called a base class by inheriting the features of the base class.
- Apart from inheriting the properties of the base class, an extra new feature can be added to the derived class.
- A base class is called parental class and the derived class is called a descendant class as it inherits the feature of the parental class.

Inheritance





Types of Inheritance

- Single inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Multiple inheritance
- Hybrid inheritance

C++ Inheritance Syntax or Format

```
class base_class_name
{
   ..................
};

class derived_class_name : acess_specifier base_class_name
{
   ...........
} ;
```

Access Specifiers

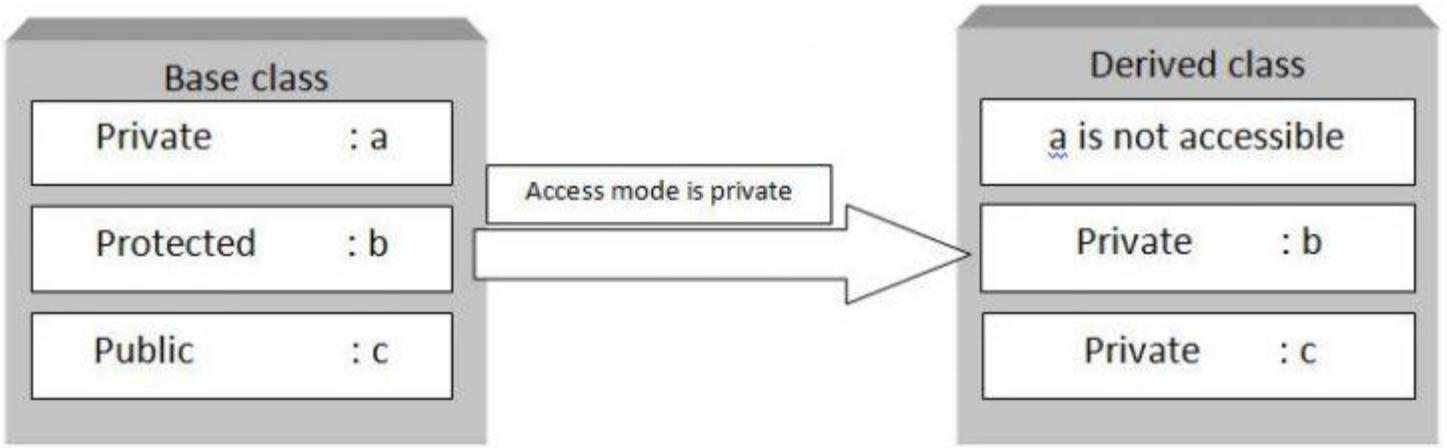
- C++ access specifiers are used for determining or setting the boundary for the availability of class members (data members and member functions) beyond that class.
- For example, the class members are grouped into sections, private, protected and public. These keywords are called access specifiers which define the accessibility or visibility level of class members.
- By default the class members are private. So if the visibility labels are missing then by default all the class members are private.

- *In inheritance, it is important to know when a member function in the base class can be used by the objects of the derived class. This is called accessibility and the access specifiers are used to determine this*

Private Access Specifiers

- If private access specifier is used while creating a class, then the public and protected data members of the base class become the private member of the derived class and private member of base class remains private.

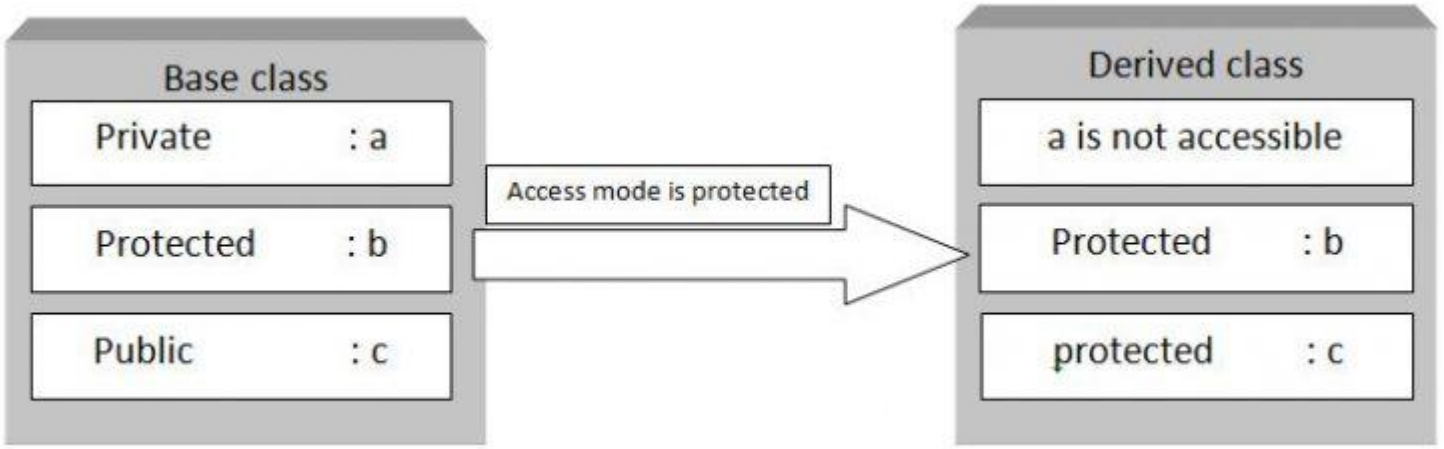
- *In this case, the members of the base class can be used only within the derived class and cannot be accessed through the object of derived class whereas they can be accessed by creating a function in the derived class.*



Protected Access Specifiers

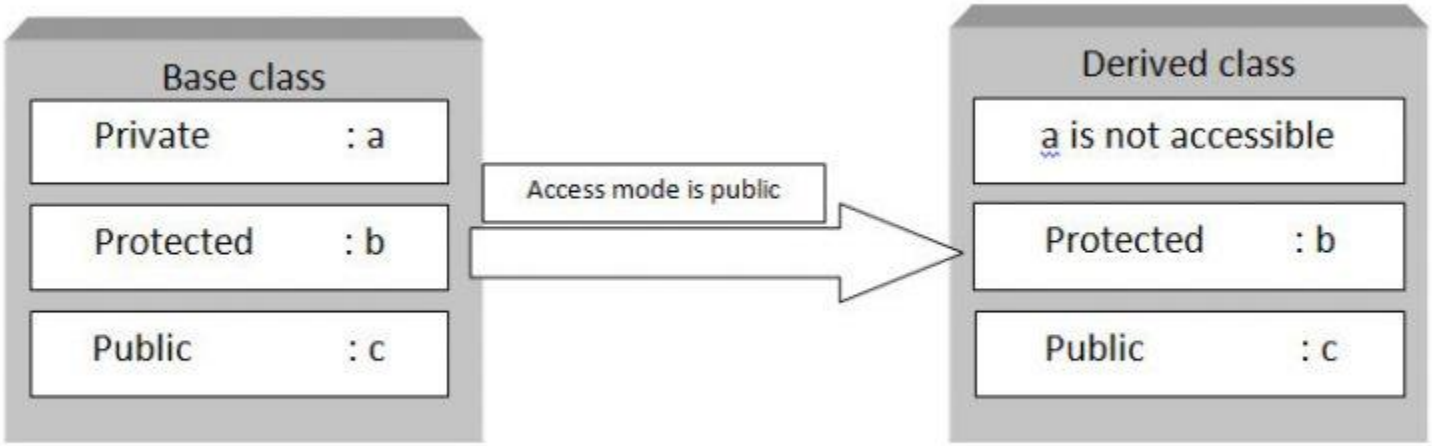
- If protected access specifier is used while deriving class then the public and protected data members of the base class becomes the protected member of the derived class and private member of the base class are inaccessible.

- In this case, the members of the base class can be used only within the derived class as protected members except for the private members.



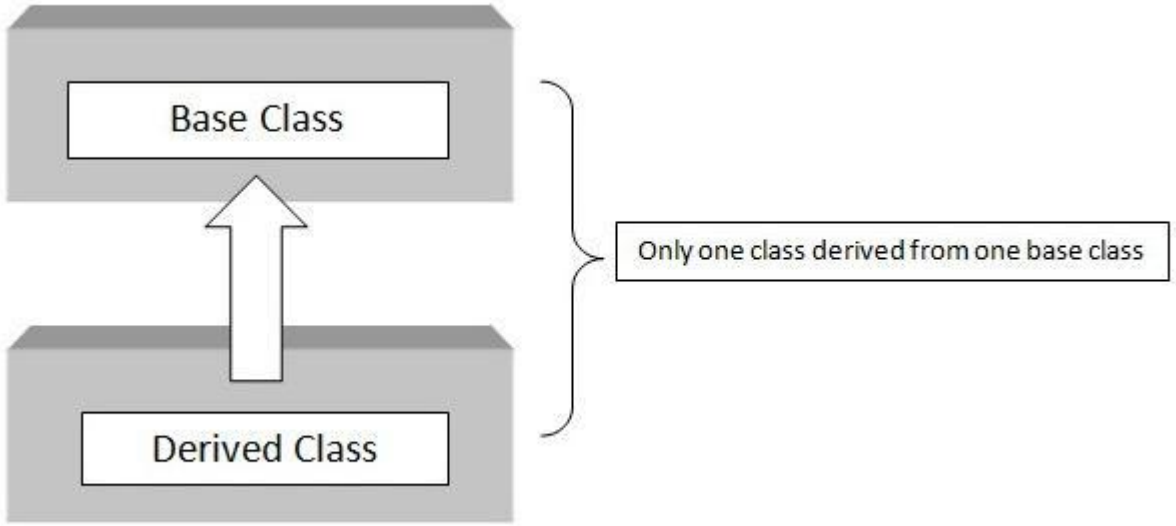
Public Access Specifiers

- If public access specifier is used while deriving class then the public data members of the base class becomes the public member of the derived class and protected members becomes the protected in the derived class but the private members of the base class are inaccessible.



Single level Inheritance

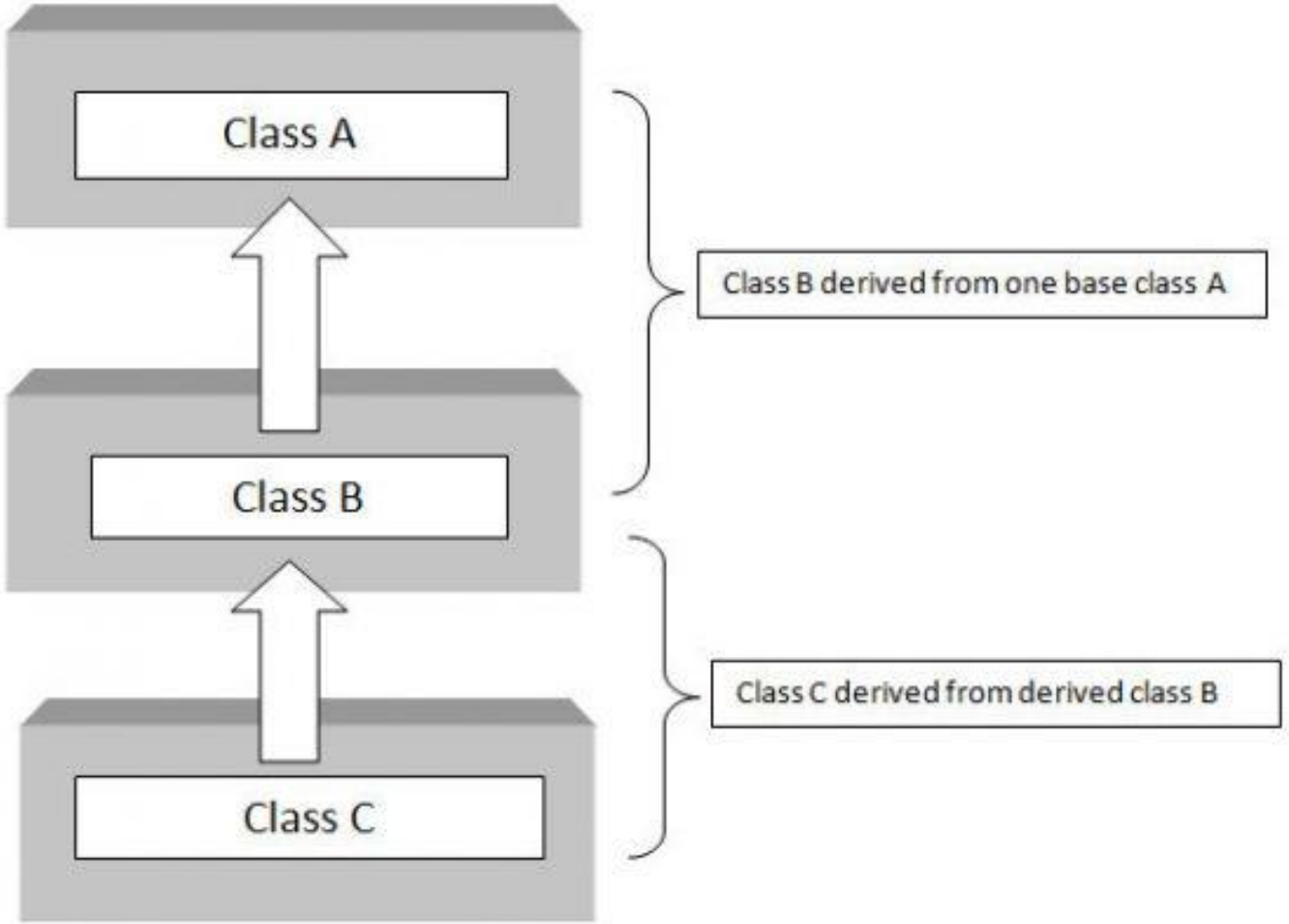
- Single inheritance only one class can be derived from the base class.
- Based on the visibility mode used or access specifier used while deriving, the properties of the base class are derived.
- Access specifier can be private, protected or public.



Multi level Inheritance

- If a class is derived from another derived class then it is called multilevel inheritance.
- So in C++ multilevel inheritance, *a class has more than one parent class.*
- For example, if we take animals as a base class then mammals are the derived class which has features of animals and then humans are the also derived class that is derived from sub-class mammals which inherit all the features of mammals.

Multi level Inheritance



Multi level Inheritance

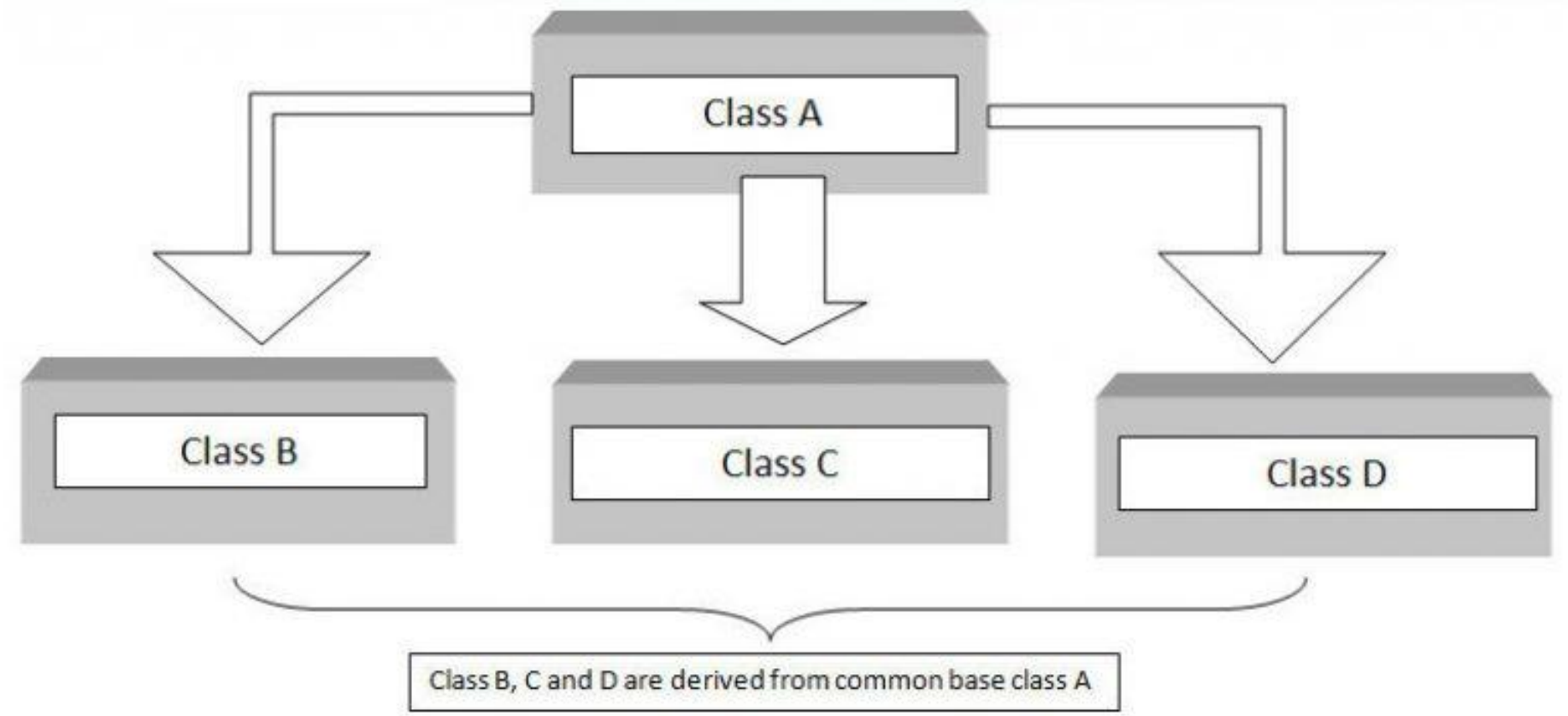
C++ Programming Multilevel Inheritance Syntax

```
class A // base class
{
     ...........
};
class B : access_specifier A // derived class
{
     ...........
 } ;
 class C : access_specifier B // derived from derived class
B
 {
     ...........
 } ;
```

Hierarchical Inheritance

- When several classes are derived from common base class it is called hierarchical inheritance.
- In C++ hierarchical inheritance, the feature of the base class is inherited onto more than one sub-class.
- For example, a car is a common class from which Audi, Ferrari, Maruti etc. can be derived.

Hierarchical Inheritance



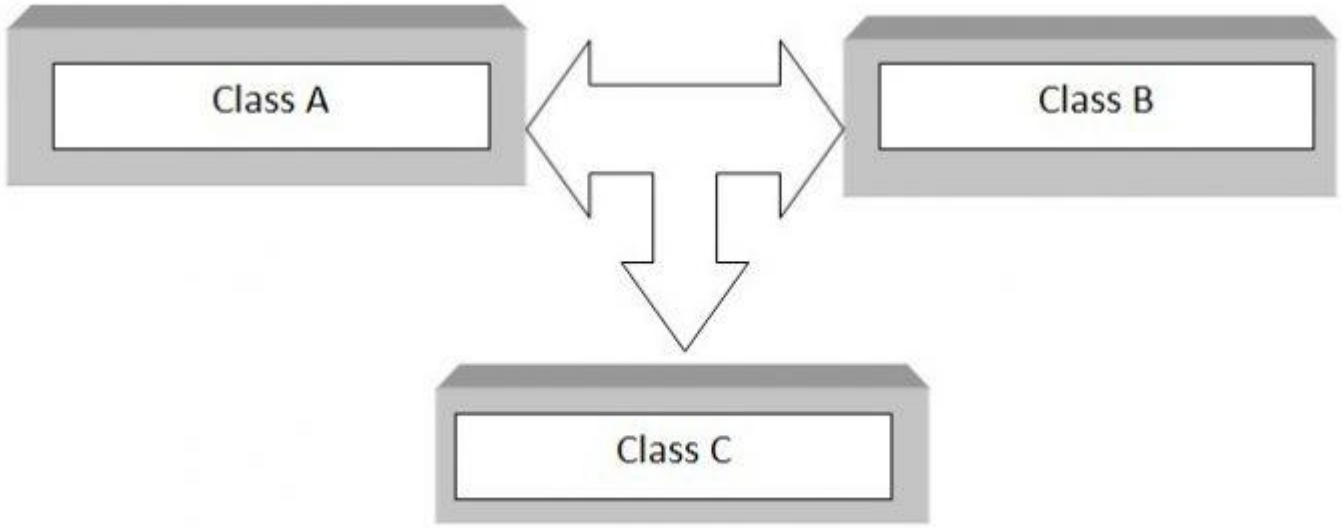
Multiple Inheritance

- If a class is derived from two or more base classes then it is called multiple inheritance.
- In C++ multiple inheritance a derived class has more than one base class.
- *How does multiple inheritance differ from multilevel inheritance?*
- Though but multiple and multilevel sounds like same but they differ hugely in meaning. In multilevel inheritance, we have multiple parent classes whereas in in multiple inheritance we have multiple base classes.

- To put it in simple words, in multilevel inheritance, a class is derived from a class which is also derived from another base class. And these levels of inheritance can be extended. On the contrary, in multiple inheritance, a class is derived from two different base classes.

Multiple Inheritance

- For example
- Multilevel inheritance : Inheritance of characters by a child from father and father inheriting characters from his father (grandfather)
- Multiple inheritance : Inheritance of characters by a child from mother and father



THANK YOU