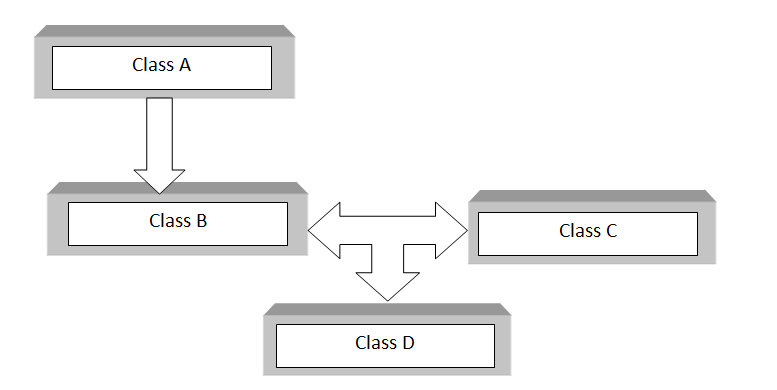
C++ Hybrid Inheritance

The inheritance in which the derivation of a class involves more than one form of any inheritance is called **hybrid inheritance**. Basically **C++ hybrid inheritance** is combination of two or more types of inheritance. It can also be called multi path inheritance.

Following block diagram highlights the concept of hybrid inheritance which involves single and multiple inheritance.

## **C++ Hybrid Inheritance Block Diagram**



Above block diagram shows the hybrid combination of single inheritance and multiple inheritance. Hybrid inheritance is used in a situation where we need to apply more than one inheritance in a program.

As in other inheritance, 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.

Click here to learn in detail about access specifiers and their use in inheritance

## **C++ Hybrid Inheritance Syntax**

class A

{

.........

};

class B : public A

{

..........

} ;

class C

{

...........

};

class D : public B, public C

{

...........

};

As shown in block diagram class B is derived from class A which is single inheritance and then Class D is inherited from B and class C which is multiple inheritance. So single inheritance and multiple inheritance jointly results in **hybrid inheritance**.

## **C++ Hybrid Inheritance Example**

Here is a simple program to illustrate the concept of hybrid inheritance in C++.

// hybrid inheritance.cpp

#include <iostream>

using namespace std;

class A

{

public:

int x;

};

class B : public A

{

public:

B() //constructor to initialize x in base class A

{

x = 10;

}

};

class C

{

public:

int y;

C() //constructor to initialize y

{

y = 4;

}

};

class D : public B, public C //D is derived from class B and class C

{

public:

void sum()

{

cout << "Sum= " << x + y;

}

};

int main()

{

D obj1; //object of derived class D

obj1.sum();

return 0;

} //end of program

**Output**

Sum= 14