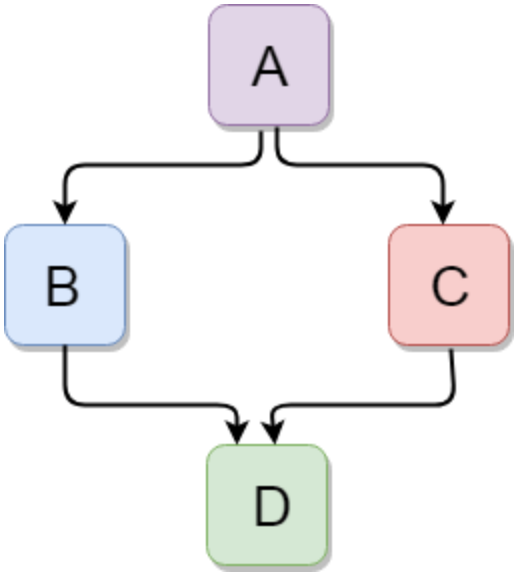
➢ C++ Hybrid Inheritance

Hybrid inheritance is a combination of more than one type of inheritance.



Let's see a simple example:

```
#include <iostream>
using namespace std;
class A
{
   protected:
   int a;
   public:
   void get_a()
   {
     std::cout << "Enter the value of 'a' : " << std::endl;
     cin>>a;
   }
};

class B : public A
{
   protected:
   int b;
   public:
   void get_b()
   {
      std::cout << "Enter the value of 'b' : " << std::endl;
     cin>>b;
```

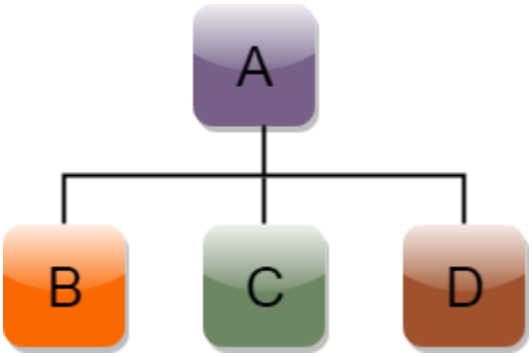
```
    }
};
class C
{
    protected:
    int c;
    public:
    void get_c()
    {
        std::cout << "Enter the value of c is : " << std::endl;
        cin>>c;
    }
};

class D : public B, public C
{
    protected:
    int d;
    public:
    void mul()
    {
        get_a();
        get_b();
        get_c();
        std::cout << "Multiplication of a,b,c is : " <<a*b*c<< std::endl;
    }
};
int main()
{
    D d;
    d.mul();
    return 0;
}
```

Output: Enter the value of 'a' :
10
Enter the value of 'b' :
20
Enter the value of c is :
30
Multiplication of a,b,c is : 6000

➢ C++ Hierarchical Inheritance

Hierarchical inheritance is defined as the process of deriving more than one class from a base class.



Syntax of Hierarchical inheritance:

```
class A
{
   // body of the class A.
}
class B : public A
{
   // body of class B.
}
class C : public A
{
   // body of class C.
}
class D : public A
{
   // body of class D.
}
```

Let's see a simple example:

```
#include <iostream>
using namespace std;
class Shape                // Declaration of base class.
{
        public:
                int a;
                int b;
        void get_data(int n,int m)
        {
                a= n;
                b = m;
```

```
        }
};
class Rectangle : public Shape  // inheriting Shape class
{
public:
        int rect_area()
        {
                int result = a*b;
                return result;
        }
};

class Triangle : public Shape    // inheriting Shape class
{
public:
        int triangle_area()
        {
                float result = 0.5*a*b;
                return result;
        }
};

int main()
{
Rectangle r;
Triangle t;
int length,breadth,base,height;

std::cout << "Enter the length and breadth of a rectangle: " << std::endl;
cin>>length>>breadth;

r.get_data(length,breadth);
int m = r.rect_area();

std::cout << "Area of the rectangle is : " <<m<< std::endl;
std::cout << "Enter the base and height of the triangle: " << std::endl;

cin>>base>>height;

t.get_data(base,height);
float n = t.triangle_area();

std::cout <<"Area of the triangle is : "  << n<<std::endl;
return 0;
}
```

Output:

```
Enter the length and breadth of a rectangle:
23
20
Area of the rectangle is : 460
Enter the base and height of the triangle:
2
5
Area of the triangle is : 5
```