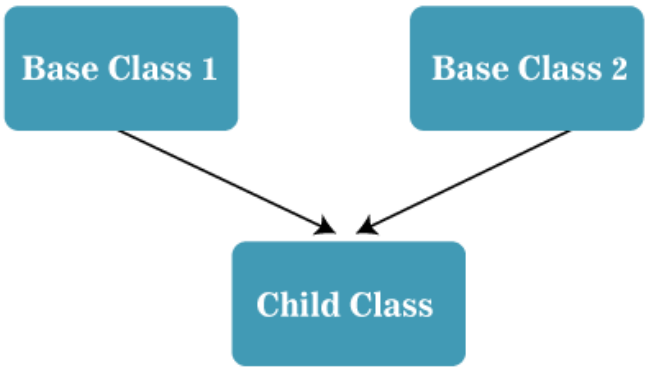
Multiple Inheritance

Multiple Inheritance is the concept of the Inheritance in C++ that **allows a child class to inherit properties or behaviour from multiple base classes**. Therefore, we can say it is the process that enables a derived class to acquire member functions, properties, characteristics from more than one base class.



Syntax of the Multiple Inheritance

```
1. class A
2. {
3. // code of class A
4. }
5. class B
6. {
7. // code of class B
8. }
9. class C: public A, public B (access modifier class_name)
10. {
11. // code of the derived class
12. }
```

In the above syntax, class A and class B are two base classes, and class C is the child class that inherits some features of the parent classes.

Let's write the various program of Multiple Inheritance to inherit the member functions and functionality from more than one base class using the derived class.

Example 1: Program to use the Multiple Inheritance

Program1.cpp

```
#include <iostream>
using namespace std;

// create a base class1
class Base_class
{
   // access specifier
   public:
   // It is a member function
   void display()
   {
```

```
        cout << " It is the first function of the Base class " << endl;
    }
};

// create a base class2
class Base_class2
{
    // access specifier
    public:
    // It is a member function
    void display2()
    {
        cout << " It is the second function of the Base class " << endl;
    }
};

/* create a child_class to inherit features of Base_class and Base_class2 with access specifier. */
class child_class: public Base_class, public Base_class2
{

    // access specifier
    public:
    void display3() // It is a member function of derive class
    {
        cout << " It is the function of the derived class " << endl;
    }

};

int main ()
{
    // create an object for derived class
    child_class ch;
    ch.display(); // call member function of Base_class1
    ch.display2(); // call member function of Base_class2
    ch.display3(); // call member function of child_class
}
```

Output

```
It is the first function of the Base class
 It is the second function of the Base class
 It is the function of the derived class
```

In the above program, we created two base classes and one child class. The **child_class** invoke member function display() and display2() from both parent classes **Base_class** and **Base_class2** with the help of child class's object ch.