**Base class and derived class constructor, implicit call and explicit call**

There are two ways to call the constructor of the base class through the constructor of the derived class, implicit and explicit. The so-called implicit method is to not specify the corresponding base class constructor in the derived class constructor. At this time, the default constructor of the base class is called (that is, the constructor with default parameter values ​​or without parameters). The so-called explicit method is to specify the constructor of the base class to be called in the constructor of the derived class, and pass part of the parameter values ​​of the derived class constructor to the base class constructor. Note: Unless the base class has a default constructor, it must be explicitly called.

**Implicit method Example**

class A

{

public:

A (int x = 0, int y = 0)

{

a = x;



b = y;

}

private:

int a;

int b;

};

//The base class A has a default constructor, which can be called implicitly

class B: public A

{

public:

B(int z = 0)

{

c = z;

}

private:

int c;

};

int main()

{

B b1;

return 0;

}

**Explicit method Example**

class A

{

public:

A(int x,int y)

{

a = x;

b = y;

}

private:

int a;

int b;

};

//The base class A does not have a default constructor, and its existing constructor needs to pass parameters through

//When the derived class constructor calls the A constructor, it must be explicitly called as follows

class B: public A

{

public:

B(int x,int y,int z):A(x,y)

{

c = z;

}

private:

int c;

};

int main ()

{

B b1(1,2,3);

return 0;

}

**Base Class Constructor Calling from Derived Class**

When we inherit class into another class then object of base class is initialized first. If a class do not have any constructor, then default constructor will be called. But if we have created any parameterized constructor then we have to initialize base class constructor from derived class.

We have to call constructor from another constructor. It is also known as constructor chaining.

When we have to call same class constructor from another constructor then we use “this” keyword. In addition, when we have to call base class constructor from derived class then we use “base” keyword.