**Content 44 (Cnt 42~part 2)**

# Constructors in Derived Class in C++

In this I had talked about which constructor will take the call first if there is Inheritance like multiple or Multilevel Inheritance.

**Important thing for constructor Calling.**

/\*

Case1:

class B: public A{

   // Order of execution of constructor -> first A() then B()

};

Case2:

class A: public B, public C{

    // Order of execution of constructor -> B() then C() and A()

};

Case3:

class A: public B, virtual public C{

    // Order of execution of constructor -> C() then B() and A()

};

\*/

**Code:**

#include <iostream>

using namespace std;

class Base1

{

    int a;

public:

    Base1(int n1)

    {

        a = n1;

        cout << "Constructor of Base1...." << endl;

    }

    void Display1()

    {

        cout << "The value of a by Base1 is: " << a << endl;

    }

};

class Base2

{

    int b;

public:

    Base2(int n2)

    {

        b = n2;

        cout << "Constructor of Base2...." << endl;

    }

    void Display()

    {

        cout << "The value of a by Base2 is: " << b << endl;

    }

};

class Derived : public Base1, public Base2 //if you write base2 first then it will call base2 first

{

    int c, d;

public:

    Derived(int x, int y, int a1, int a2) : Base1(x), Base2(y) //change here does'nt matter

    // like you write base2 first and then base1

    {

        c = a1;

        d = a2;

        cout << "The derived class constructor is called..."<<endl;

    }

    void show()

    {

        cout << "The value of c is: " << c << endl;

        cout << "The value of d is: " << d << endl;

    }

};

int main()

{

    Derived d(2,4,6,8);

    d.Display1();

    d.Display();

    d.show();

    return 0;

}

**Ouput:**

Constructor of Base1....

Constructor of Base2....

The derived class constructor is called...

The value of a by Base1 is: 2

The value of a by Base2 is: 4

The value of c is: 6

The value of d is: 8