构造函数和析构函数调用顺序

#include<iostream.h>

class A

{

public:

A() { cout << "A is created!" << endl; }

~A() { cout << "A is destroyed!" << endl; }

};

class B

{

public:

B() { cout << "B is created!" << endl; }

~B() { cout << "B is destroyed!" << endl; }

};

class C

{

public:

C() { cout << "C is created!" << endl; }

~C() { cout << "C is destroyed!" << endl; }

};

class D : public C

{

B b;

A a;

public:

D():a(), b() { cout << "D is created!" << endl; }

~D() { cout << "D is destroyed!" << endl; }

};

void main()

{

D d;

}