#include <iostream>

using namespace std;

class A {

public:

A();

A(const A& a);

A(string name);

~A();

public:

A& operator=(const A& org);

};

A::A()

{

cout << "construct null" << endl;

}

A::A(string name)

{

cout << "construct with name" << endl;

}

A::A(const A& a)

{

cout << "copy construct " << endl;

}

A::~A()

{

cout << "destructor" << endl;

}

A& A::operator=(const A& org)

{

return \*this;

}

A a;

int main()

{

a = A("teatcher"); // 执行构造函数A(string)后会立即执行析构函数

int b = 0;

A a1;

a1 = A("student"); // 执行构造函数A(string)后会立即执行析构函数

int c = 0;

return 0;

}

construct null

construct with name

destructor

construct null

construct with name

destructor

destructor

destructor

从上例中可以看出，如果一个类对象被多次构造，则在第2次及以后的构造完成后，会立刻执行析构函数。

个人理解：一个类对象在程序结束时，系统会自动调用一次析构函数来释放其当前的资源，之前调用构造函数所分配的内存空间必须在下一次构造之前就析构掉，否则会出现内存的泄露。