对象作为返回值，也会有拷贝构造

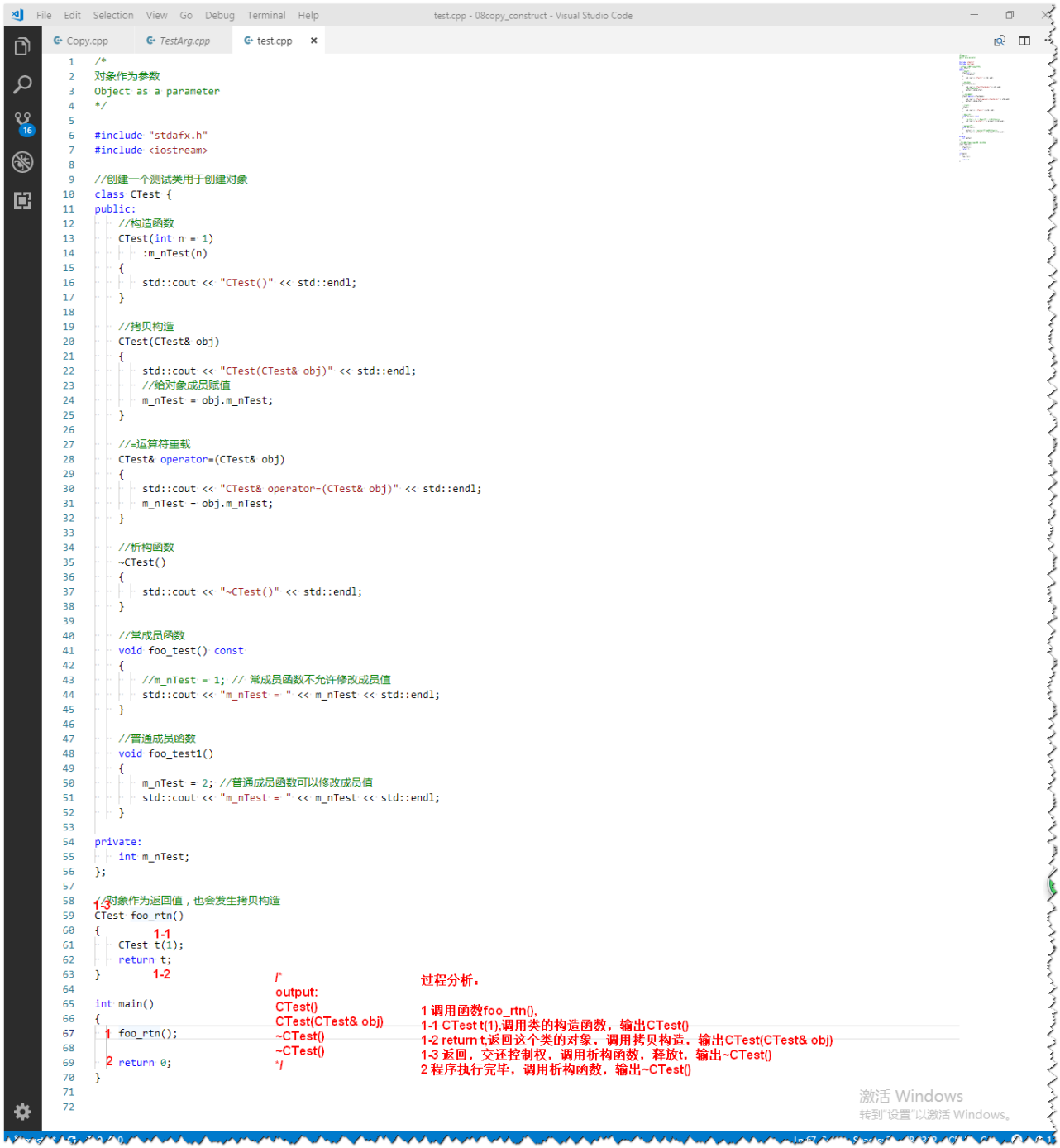
1. 如果外部没有接受函数的对象，该返回的对象是个无名对象，当遇到该行的“;”时，析构。

```
foo_rtn().foo_test() , printf("123\r\n");printf("456\r\n");
123
~CTest() -->说明遇到；析构
456
```

```
/*
对象作为参数
Object as a parameter
*/

#include "stdafx.h"
#include <iostream>

//创建一个测试类用于创建对象
class CTest {
public:
    //构造函数
    CTest(int n = 1)
        :m_nTest(n)
    {
        std::cout << "CTest()" << std::endl;
    }

    //拷贝构造
    CTest(CTest& obj)
    {
        std::cout << "CTest(CTest& obj)" << std::endl;
        //给对象成员赋值
        m_nTest = obj.m_nTest;
    }

    //=运算符重载
    CTest& operator=(CTest& obj)
```

```
    {
        std::cout << "CTest& operator=(CTest& obj)" << std::endl;
        m_nTest = obj.m_nTest;
    }

    //析构函数
    ~CTest()
    {
        std::cout << "~CTest()" << std::endl;
    }

    //常成员函数
    void foo_test() const
    {
        //m_nTest = 1; // 常成员函数不允许修改成员值
        std::cout << "m_nTest = " << m_nTest << std::endl;
    }

    //普通成员函数
    void foo_test1()
    {
        m_nTest = 2; //普通成员函数可以修改成员值
        std::cout << "m_nTest = " << m_nTest << std::endl;
    }

private:
    int m_nTest;
};

//对象作为返回值，也会发生拷贝构造
CTest foo_rtn()
{
    CTest t(1);
    return t;
}

int main()
{
    foo_rtn();

    return 0;
}
```

```
71
72
```


File Edit Selection View Go Debug Terminal Help
test.cpp - 08copy_construct - Visual Studio Code
Copy.cpp
TestArg.cpp
test.cpp
/*
对象作为参数
Object as a parameter
*/
#include "stdafx.h"
#include <iostream>
//创建一个测试类用于创建对象
class CTest {
public:
//构造函数
CTest(int n = 1)
:m_nTest(n)
{
std::cout << "CTest()" << std::endl;
}
//拷贝构造
CTest(CTest& obj)
{
std::cout << "CTest(CTest& obj)" << std::endl;
//给对象成员赋值
m_nTest = obj.m_nTest;
}
//=运算符重载
CTest& operator=(CTest& obj)
{
std::cout << "CTest& operator=(CTest& obj)" << std::endl;
m_nTest = obj.m_nTest;
}
//析构函数
~CTest()
{
std::cout << "~CTest()" << std::endl;
}
//常成员函数
void foo_test() const
{
//m_nTest = 1; // 常成员函数不允许修改成员值
std::cout << "m_nTest = " << m_nTest << std::endl;
}
//普通成员函数
void foo_test1()
{
m_nTest = 2; //普通成员函数可以修改成员值
std::cout << "m_nTest = " << m_nTest << std::endl;
}
private:
int m_nTest;
};
//对象作为返回值，也会发生拷贝构造
CTest foo_rtn()
{
CTest t(1);
return t;
}
int main()
{
foo_rtn();
return 0;
}
1-3
1-1
1-2
1
2
/*
output:
CTest()
CTest(CTest& obj)
~CTest()
~CTest()
*/
过程分析：
1 调用函数foo_rtn(),
1-1 CTest t(1),调用类的构造函数，输出CTest()
1-2 return t,返回这个类的对象，调用拷贝构造，输出CTest(CTest& obj)
1-3 返回，交还控制权，调用析构函数，释放t，输出~CTest()
2 程序执行完毕，调用析构函数，输出~CTest()
激活 Windows
转到"设置"以激活 Windows。