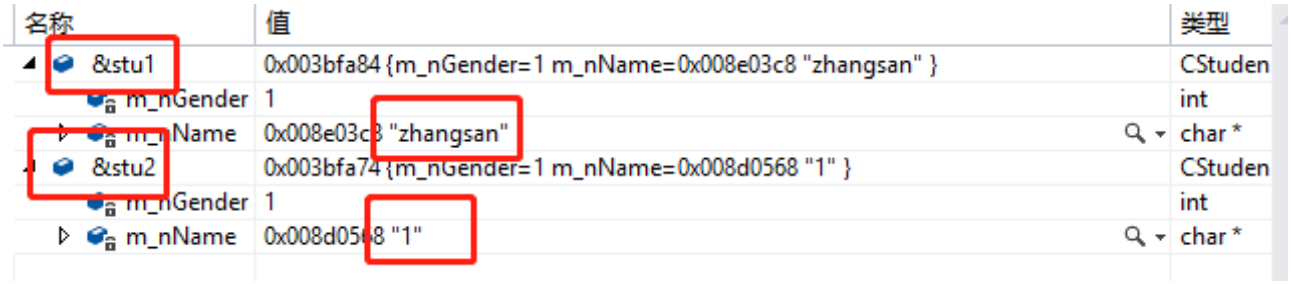
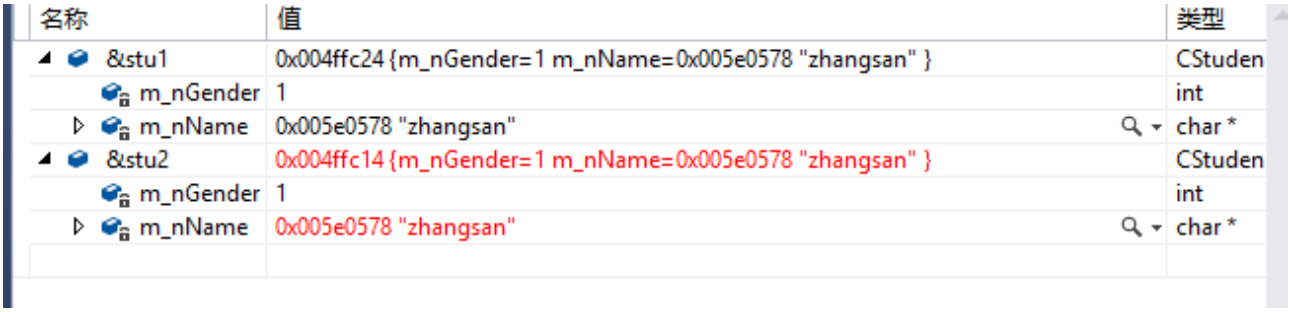
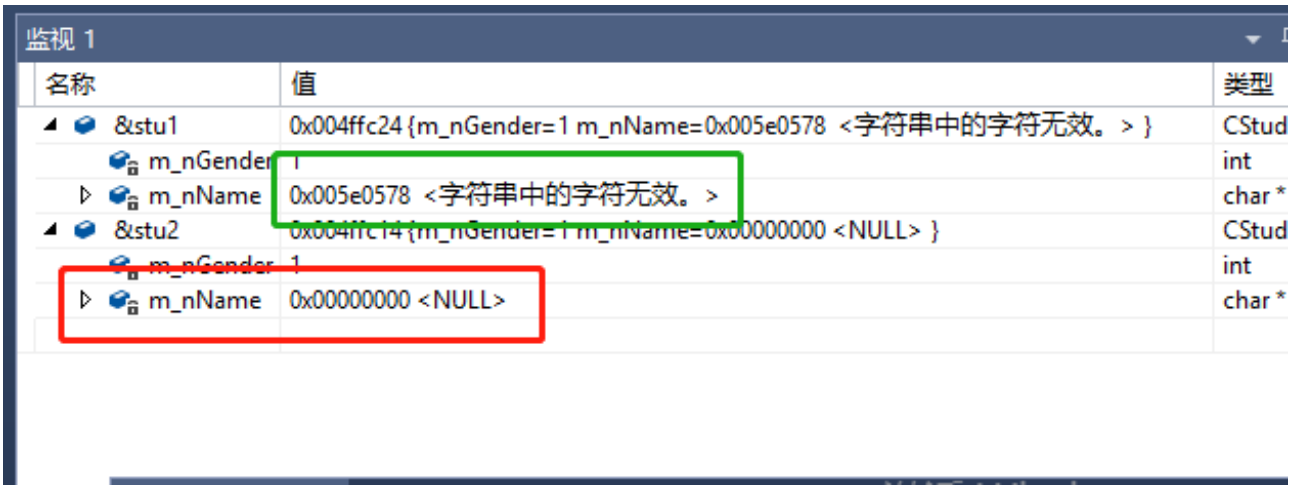
浅拷贝



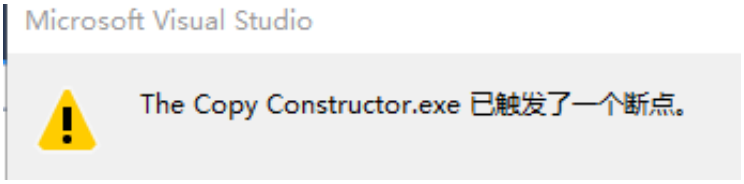
stu2拷贝给stu1



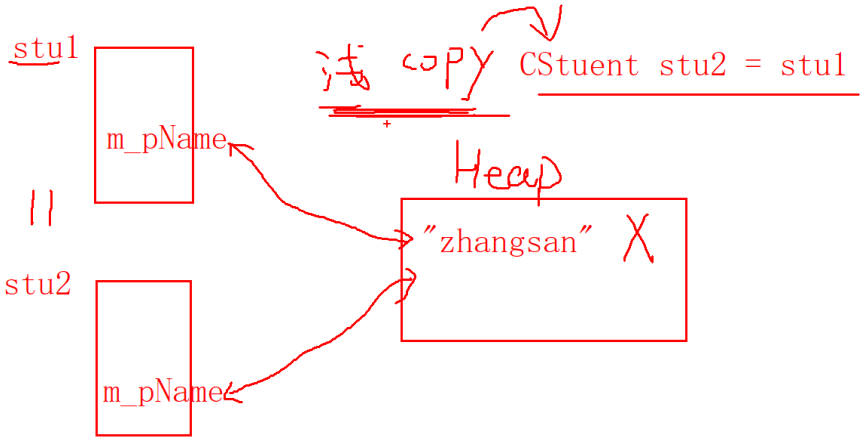
程序执行完毕，开始析构，把stu2申请的对象析构了，
但是同时stu1申请的空间内存中的参数也被影响了，
变成了无效的参数。这就造成了浅拷贝。



触发异常断点



浅拷贝图示




```cpp
/*
浅拷贝
*/

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

class CStudent {
public:
    //构造函数
    CStudent(char* pName, int nGender = 1)
        :m_nGender(nGender) //初始化参数列表
    {
        //申请堆空间，保存姓名
        m_nName = new char[strlen(pName) + 1];
        //拷贝姓名
        strcpy_s(m_nName, strlen(pName) + 1, pName);
        std::cout << "CStudent(int nGender)" << std::endl;
    }
    //析构函数
    ~CStudent()
    {
        std::cout << "~CStudent()" << std::endl;
        //释放申请的堆空间
        if (m_nName != NULL) {
            delete[] m_nName;
        }
        m_nName = NULL;
```

```
    }

private:
    int m_nGender; //性别
    char* m_nName; //姓名
};

int main()
{
    char szName[] = "zhangsan";
    //申请一个CStudent类对象，名为stu1
    //调用CStudent类的构造函数
    CStudent stu1(szName, 1);
    //申请一个CStudent类对象，名为stu2
    //调用CStudent类的构造函数
    CStudent stu2("1");
    //将stu1对象拷贝给stu2对象
    stu2 = stu1;

    return 0;
}
```