

C++类中的成员可以是另一个类的对象，我们称该成员为** 对象成员**

```
class A {}  
class B  
{  
    A a;  
}
```

此时，B类中有一个成员a，类型为A。

下面我们需要重点讨论的就是——

当其他类作为类的成员，则当用本类创建对象时，会先生成一个其他对象并以此作为我们这个类的成员。

但是，在析构时，则是先析构我们的对象，再析构类内的对象。

总结：

构造时：**内层先，外层后**

析构时：**外层先，内层后**

在代码中的体现：

```
class Phone  
{  
public:  
    Phone(string name)  
    {  
        m_PhoneName = name;  
        cout << "Phone构造" << endl;  
    }  
    ~Phone()  
    {  
        cout << "Phone析构" << endl;  
    }  
    string m_PhoneName;  
};  
  
class Person  
{  
public:  
    //初始化列表可以告诉编译器调用哪一个构造函数  
    Person(string name, string pName) :m_Name(name), m_Phone(pName)  
    {  
        cout << "Person构造" << endl;  
    }  
    ~Person()
```

```

    {
        cout << "Person析构" << endl;
    }
void playGame()
{
    cout << m_Name << " 使用" << m_Phone.m_PhoneName << " 牌手机! " << endl;
}
string m_Name;
Phone m_Phone;

};
int main()
{
    Person p("小王", "大哥大");
    p.playGame();
    system("pause");
    return 0;
}

```

输出：

Phone构造

Person构造

小王使用大哥大牌手机

Person析构

Phone析构