

# new与delete补充

---

在没有显式给出析构函数时，

```
TestClass *ptr = new Testclass[3];  
delete ptr;
```

此时不会崩溃，原因在于在堆结构上上溢标志后没有记录对象的个数。而显式给出析构函数后，则会保存对象个数。在基本数据类型上，也不会保存对象的个数。

## 拷贝构造

---

### 拷贝构造的用法

---

- 当使用一个已经存在的对象创建新对象的时候，C++会调用拷贝构造

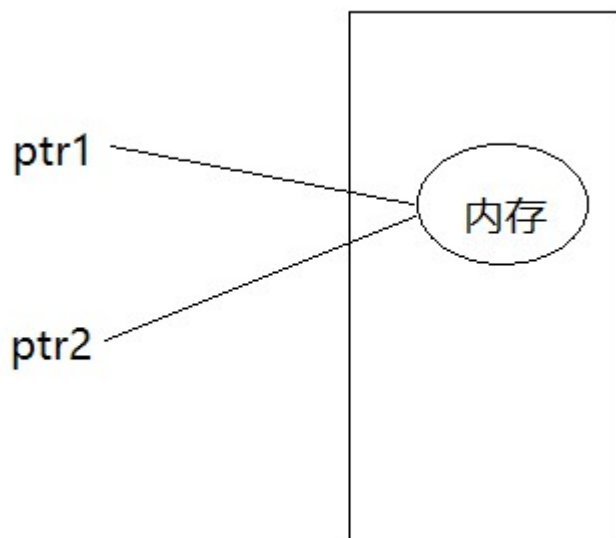
```
class Testclass {  
    ...  
    // 拷贝构造  
    Testclass(Testclass &obj)  
    {  
        ...  
    }  
    ...  
};
```

- 当不提供拷贝构造时，编译器会提供一个默认拷贝构造——值拷贝。

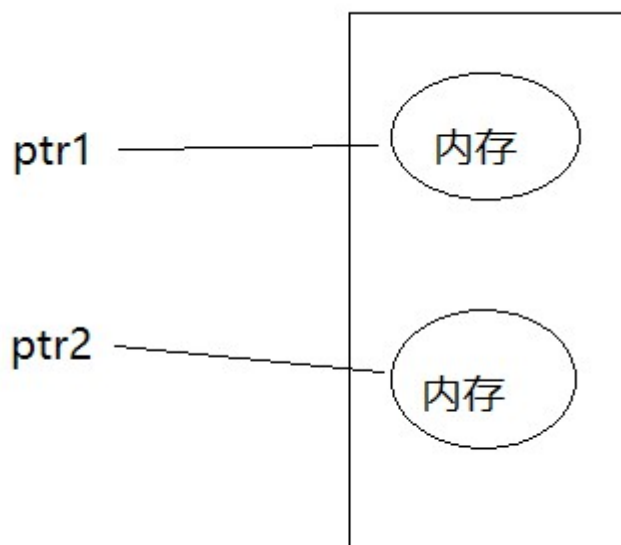
### 浅拷贝与深拷贝

---

- 浅拷贝



- 深拷贝



## 拷贝构造的时机

### 1. 对象传参

- 当函数被调用的时候，拷贝构造被调用
- 当出函数作用域的时候，析构被调用

### 2. 对象初始化对象

```
Testclass t1;  
Testclass t2(t1);  
Testclass t3 = t1;
```

### 3. 对象作为返回值

- 碰到分号释放

#### 4. 无名对象

```
TestClass().foo();  
TestClass("Hi").foo();
```