### 逐位复制(bitwise copy)

* 1. 问题：对象成员进行逐位复制的结果是拥有两个一模一样的实例，而这个副本里面的同名指针会指向相同的地址，于是如果删除其中一个对象，包含的指针也会被删除，万一此时另一个副本还在引用这个指针，就会报错。

### 重载操作符

MyClass &operator=(const MyClass &rhs);

const引用声明常量 就不怕影响到副本

MyClass &myClass::operator=(const MyClass \*chs)

{

if(this != &chs)

{

delete ptr;

ptr = new int;

\*ptr = \*rhs.ptr;

}

else

{

std::cout<<”赋值号两边为同一个对象，不做处理！\n”;

}

return \*this;

}

### 副本构造器

问题：

MyClass obj1;

MyClass obj2=obj1;

先创建一个实例，再在创建实例obj2的同时用obj1的值对它进行赋值。

这时候编译器会寻找一个副本构造器(copy constructor)，如果找不到，就会自行创建一个。

这样即使重载了操作符，但是“逐位复制”还是会发生。

MyClass(const MyClass &chs);//自定义一个副本构造器