

赋值运算符复制构造函数（拷贝构造函数）易混处总结计算机等级考试 PDF 转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/645/2021_2022__E8_B5_8B_E5_80_BC_E8_BF_90_E7_c97_645574.htm 赋值运算符和复制构造函数都是用已存在的 B 对象来创建另一个对象 A。不同之处在于：赋值运算符处理两个已有对象，即赋值前 B 应该是存在的；复制构造函数是生成一个全新的对象，即调用复制构造函数之前 A 不存在。

`CTemp a(b).` // 复制构造函数，C 风格的初始化
`CTemp a=b.` // 仍然是复制构造函数，不过这种风格只是为了与 C 兼容，与上面的效果一样，在这之前 a 不存在，或者说还未构造好。
`CTemp a. a=b.` // 赋值运算符 在这之前 a 已经通过默认构造函数构造完成。

实例总结：重点：包含动态分配成员类应提供拷贝构造函数，并重载 "=" 赋值操作符。以下讨论中将用到的例子：

```
class CExample { public:
    CExample(){ pBuffer=NULL. nSize=0.} ~CExample(){ delete pBuffer.}
    void Init(int n){ pBuffer=new char[n]. nSize=n.} private:
    char *pBuffer. //类的对象中包含指针,指向动态分配的内存资源
    int nSize. }. 这个类的主要特点是包含指向其他资源的指针。
    pBuffer指向堆中分配的一段内存空间。
```

一、拷贝构造函数 调用拷贝构造函数

```
1 int main(int argc, char* argv[]) { CExample
theObjone. theObjone.Init(40). //现在需要另一个对象,需要将他
初始化称对象一的状态 CExample theObjtwo=theObjone. //拷贝
构造函数 ... } 语句"CExample theObjtwo=theObjone."
```

用 theObjone 初始化 theObjtwo。其完成方式是内存拷贝，复制所有成员的值。完成后

，`theObjtwo.pBuffer==theObjone.pBuffer`。即它们将指向同样

的地方（地址空间），指针虽然复制了，但所指向的空间内容并没有复制，而是由两个对象共用了。这样不符合要求，对象之间不独立了，并为空间的删除带来隐患。所以需要采用必要的手段来避免此类情况。回顾以下此语句的具体过程：通过拷贝构造函数（系统默认的）创建新对象theObjtwo，并没有调用theObjtwo的构造函数（vs2005试验过）。可以在自定义的拷贝构造函数中添加输出的语句测试。注意：对于含有在自由空间分配的成员时，要使用深度复制，不应使用浅复制。调用拷贝构造函数2 当对象直接作为参数传给函数时，函数将建立对象的临时拷贝，这个拷贝过程也将调用拷贝构造函数。例如 BOOL testfunc(CExample obj).

testfunc(theObjone). //对象直接作为参数。 BOOL

testfunc(CExample obj) { //针对obj的操作实际上是针对复制后的临时拷贝进行的 } 调用拷贝构造函数3 当函数中的局部对象被返回给函数调者时，也将建立此局部对象的一个临时拷贝，拷贝构造函数也将被调用 CTest func() { CTest theTest.

return theTest } 二、赋值符的重载 下面的代码与上例相似 int main(int argc, char* argv[]) { CExample theObjone.

theObjone.Init(40). CExample theObjthree. theObjthree.Init(60). //现在需要一个对象赋值操作,被赋值对象的原内容被清除，并用右边对象的内容填充。 theObjthree=theObjone. return 0. } 也用到了"="号，但与"一、"中的例子并不同，"一、"的例子中，"="在对象声明语句中，表示初始化。更多时候,这种初始化也可用括号表示。例如 CExample theObjone(theObjtwo). 而本例子中，"="表示赋值操作。将对象theObjone的内容复制到对象theObjthree.，这其中涉及到对象theObjthree原有内容的丢弃

，新内容的复制。但"="的缺省操作只是将成员变量的值相应复制。旧的值被自然丢弃。由于对象内包含指针，将造成不良后果:为了避免内存泄露，指针成员将释放指针所指向的空间，以便接受新的指针值，这正是由赋值运算符的特征所决定的。但如果是"x=x"即自己给自己赋值，会出现什么情况呢？x将释放分配给自己的内存，然后，从赋值运算符右边指向的内存中复制值时，发现值不见了。因此，包含动态分配成员类除提供拷贝构造函数外，还应该考虑重载"="赋值操作符号。类定义变为: class CExample { ... CExample(const CExample

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com