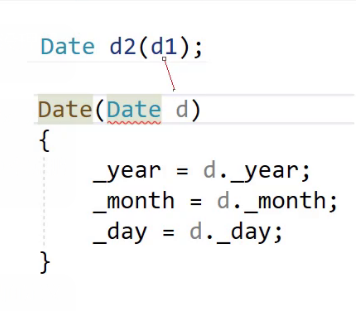
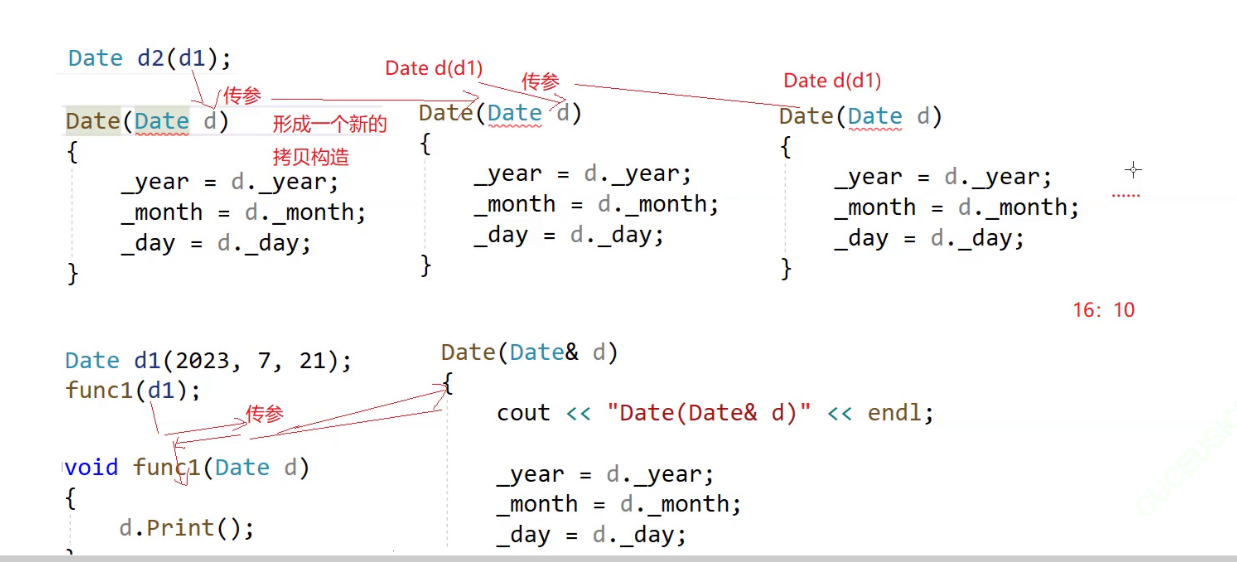
起因



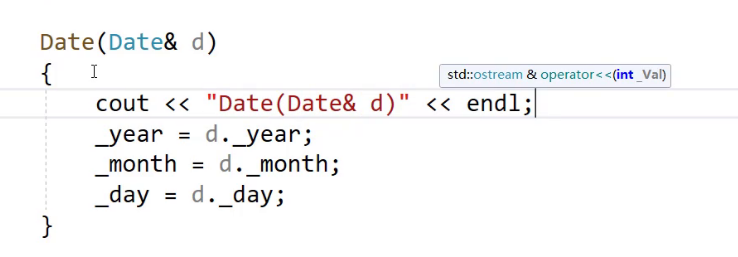
无穷地柜

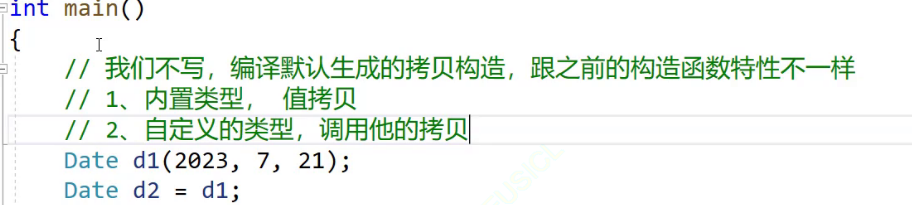


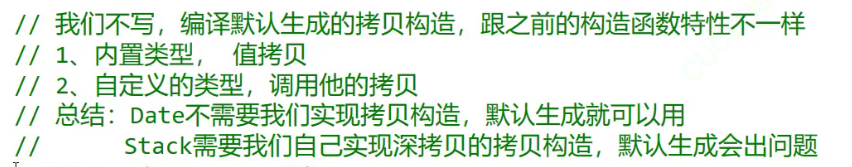
自定义函数一般需要调用拷贝函数，不鞥直接传值传参，

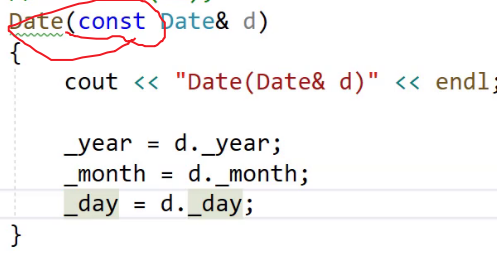
因为会出现二次析构的情况

应该改为；









**深拷贝和浅拷贝**

**拷贝构造函数**

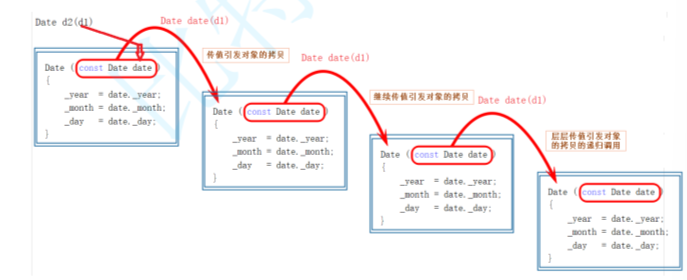
**1、概念**

只有单个形参，该形参是对本类类型对象的引用（一般常用const修饰），在用已存在的类类型对象创建新对象时**由编译器自动调用**

**2、特性**

拷贝构造函数是构造函数的一个重载形式

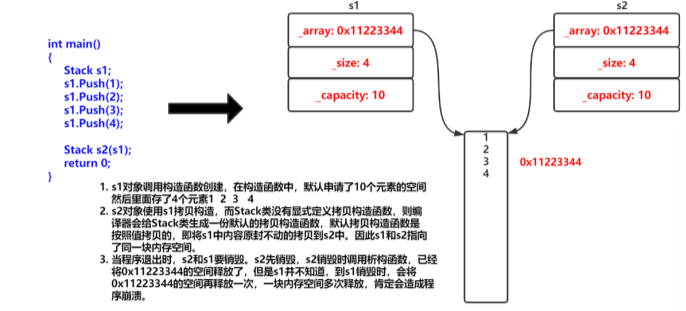
拷贝构造函数的参数只有一个且必须是类类型对象的引用，**使用传值方式编译器直接报错**，因为会引发无穷递归调用



若未显式定义，编译器会生成默认的拷贝构造函数。 默认的拷贝构造函数对象按内存存储按字节序完成拷贝，这种拷贝叫做浅拷贝，或者值拷贝

注意：在编译器生成的默认拷贝构造函数中，内置类型是按照字节方式直接拷贝的，而自定义类型是调用其拷贝构造函数完成拷贝的

编译器生成的默认拷贝构造函数已经可以完成字节序的值拷贝了，但我们还是需要自定义构造拷贝函数，不然容易出现二次释放的问题如下



注意：类中如果没有涉及资源申请时，拷贝构造函数是否写都可以；一旦涉及到资源申请时，则拷贝构造函数是一定要写的，否则就是浅拷贝

拷贝构造函数典型调用场景：

使用已存在对象创建新对象

函数参数类型为类类型对象

函数返回值类型为类类型对象

**赋值运算符重载**

**起因：**

C++为了增强代码的可读性引入了运算符重载，运算符重载是具有特殊函数名的函数，也具有其返回值类型，函数名字以及参数列表，其返回值类型与参数列表与普通的函数类似。函数名字为：关键字operator后面接需要重载的运算符符号。函数原型：返回值类型 operator操作符(参数列表)

**注意**：

不能通过连接其他符号来创建新的操作符：比如operator@

重载操作符必须有一个类类型参数

用于内置类型的运算符，其含义不能改变，例如：内置的整型+，不 能改变其含义

作为类成员函数重载时，其形参看起来比操作数数目少1，因为成员函数的第一个参数为隐藏的this

**.\* :: sizeof ?: .** 注意以上5个运算符不能重载。这个经常在笔试选择题中出现

**赋值运算符重载**

**赋值运算符重载格式**

参数类型：const T&，传递引用可以提高传参效率

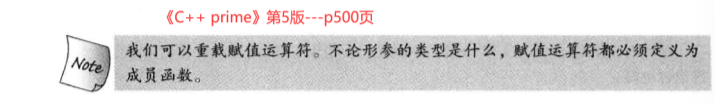
返回值类型：T&，返回引用可以提高返回的效率，有返回值目的是为了支持连续赋值

检测是否自己给自己赋值

返回\*this ：要复合连续赋值的含义

**赋值运算符只能重载成类的成员函数不能重载成全局函数**

原因：赋值运算符如果不显式实现，编译器会生成一个默认的。此时用户再在类外自己实现一个全局的赋值运算符重载，就和编译器在类中生成的默认赋值运算符重载冲突了，故赋值运算符重载只能是类的成员函数



**用户没有显式实现时，编译器会生成一个默认赋值运算符重载，以值的方式逐字节拷贝。注意：内置类型成员变量是直接赋值的，而自定义类型成员变量需要调用对应类的赋值运算符重载完成赋值。**

**前置++和后置++重载**

2023/08/20 23:42