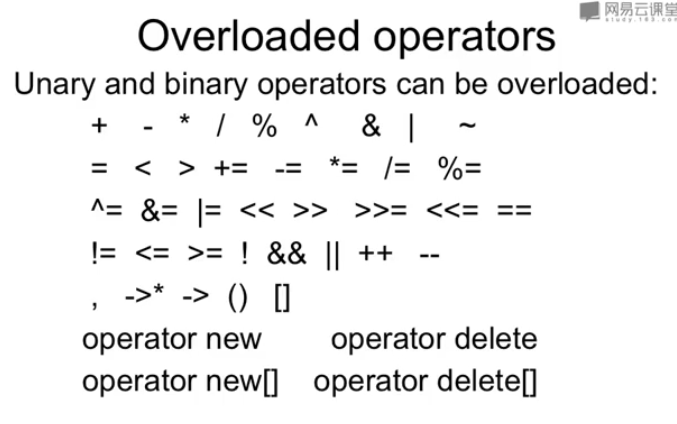
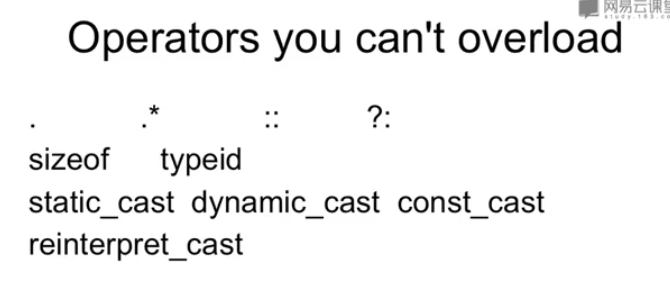
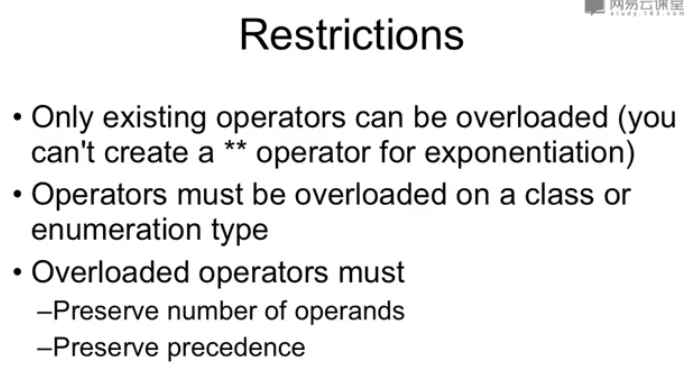


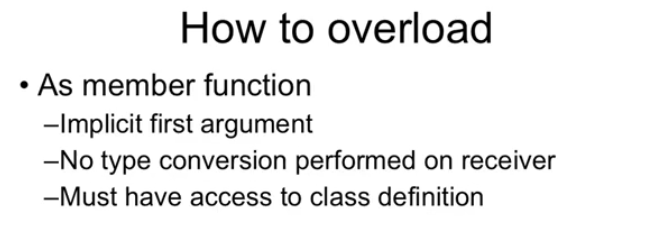
* 使得用户自定义的类型可以表现得和**内建类型**一样
* 另外一种方式**进行函数**调用





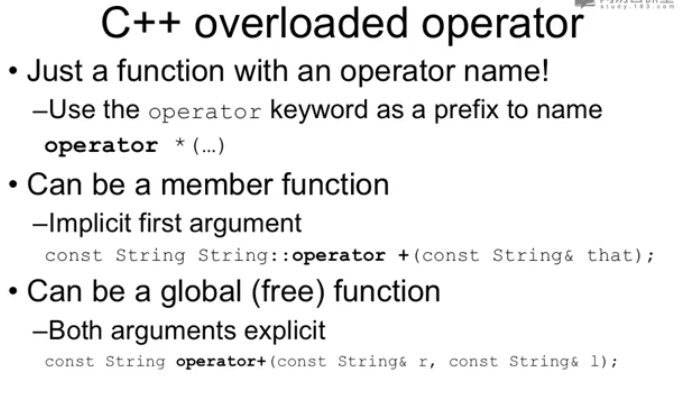


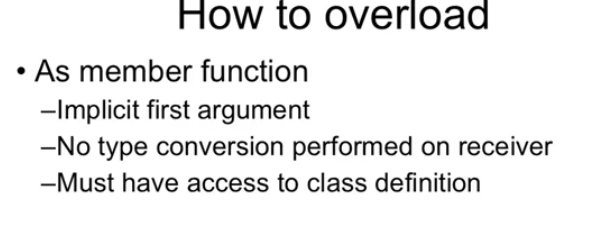
* 只能重载已有运算符
* 只能在类或枚举类型上重载

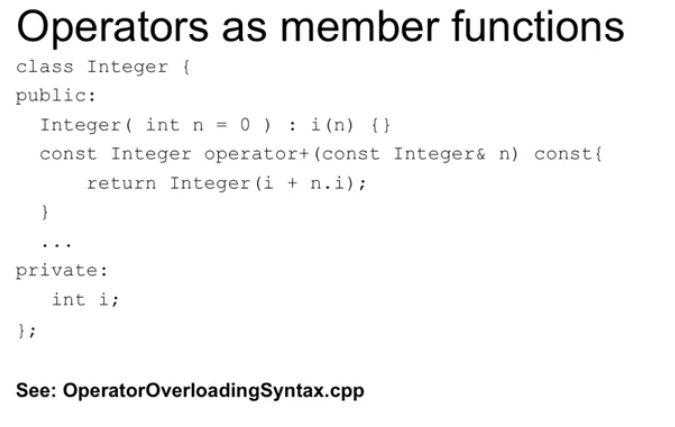


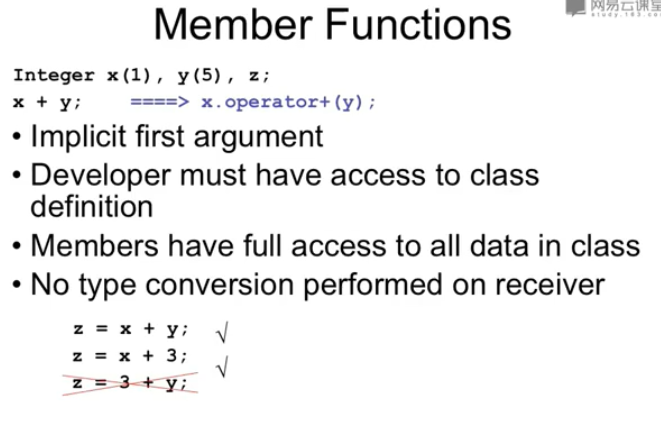
作为成员函数 进行重载

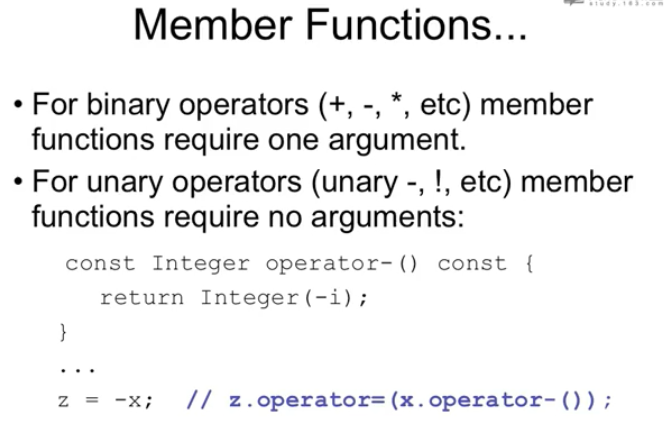
* 隐式的第一个参数
* receiver 不进行类型转化
* ?

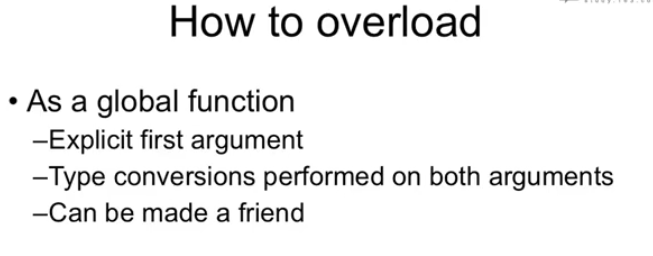




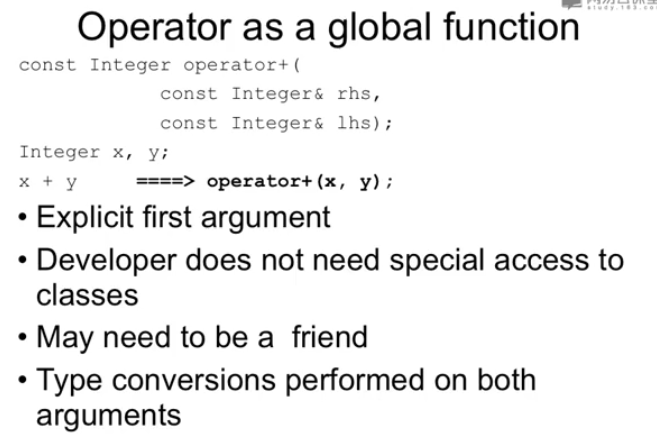


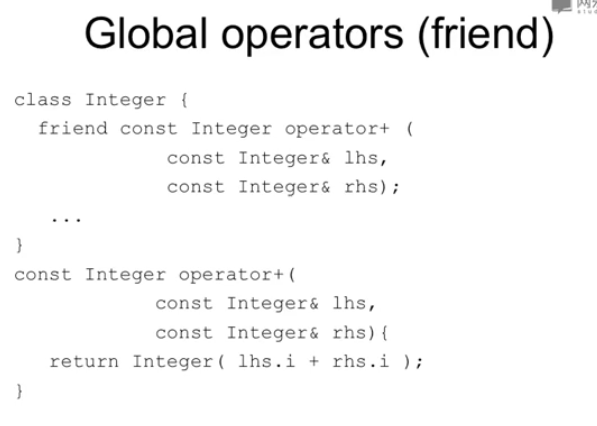


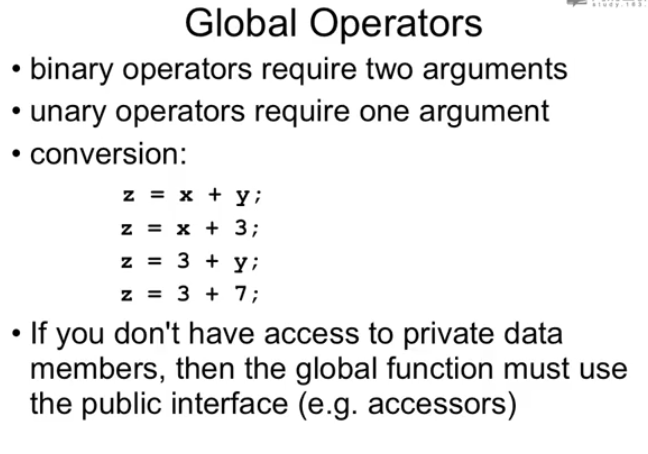


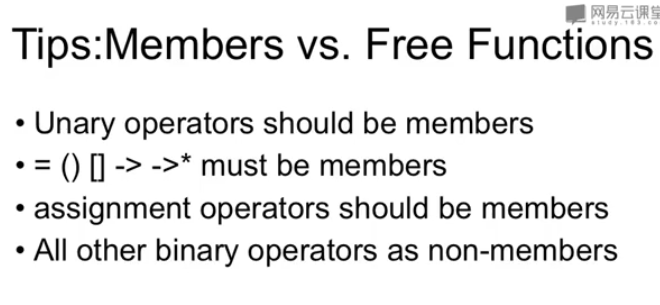


* 作为全局函数重载
  + **显示**的第一个参数
  + 两个参数上都进行 **类型转换** type conversion
  + 可以作为一个友元 friend

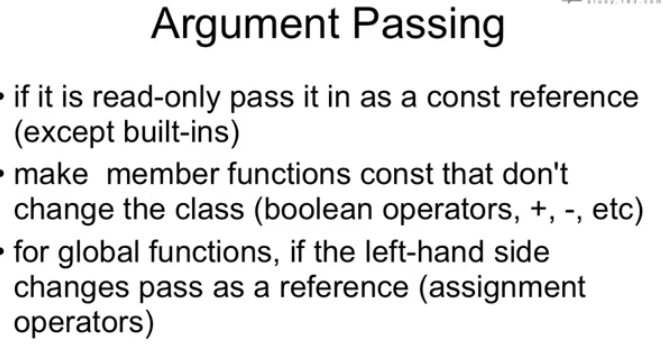






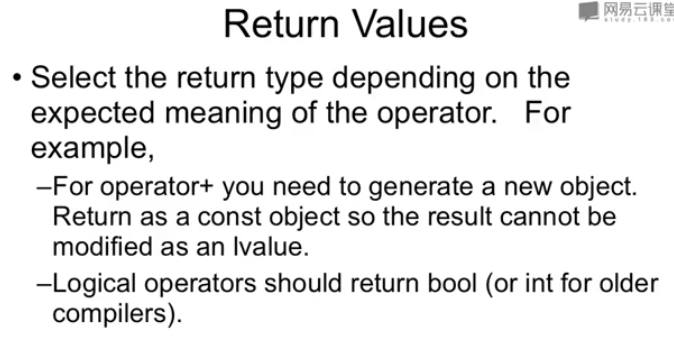


# # 31 原型



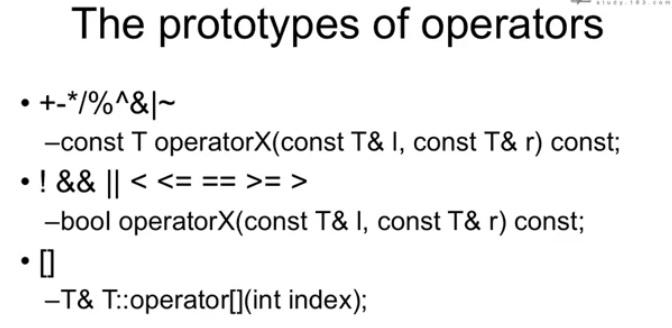
**参数传递**

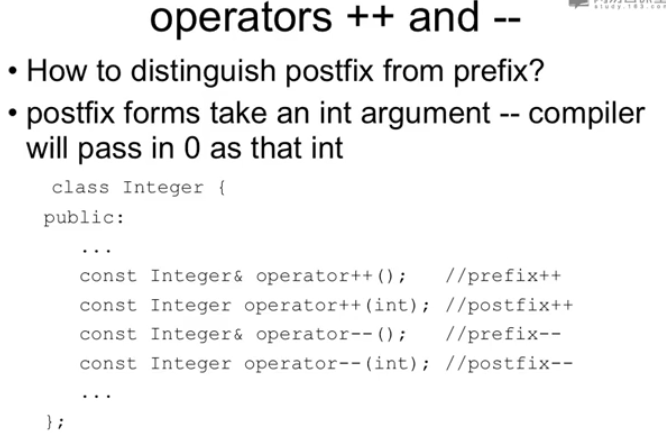
1. 如果只读的话，按照“**常引用**”的方式传参。
2. 如果不修改类， 将成员函数声明为常const的。（布尔运算，+/- 如）
3. 对于全局函数，如果会修改左侧的运算子，那么就将左侧运算子声明为**引用。（赋值运算）**



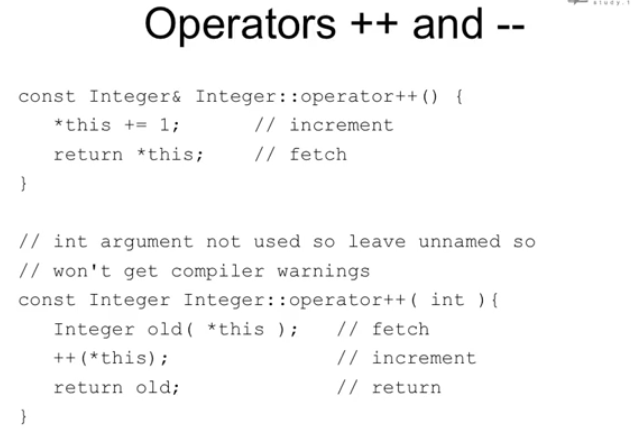
返回值

1. 返回值类型由运算的意义确定。比如逻辑运算返回布尔变量，+ 运算符返回一个新的对象

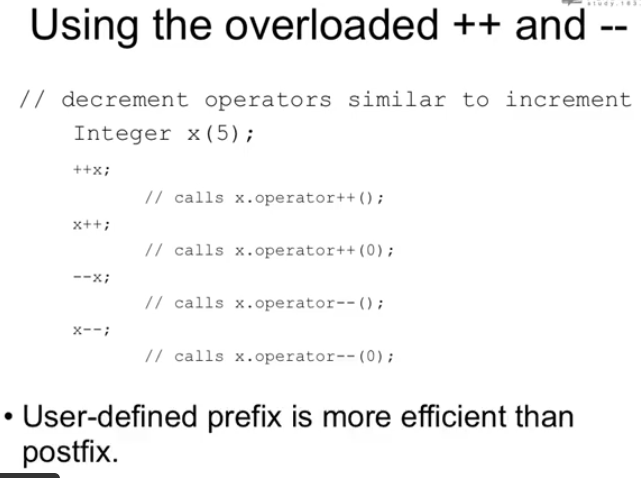


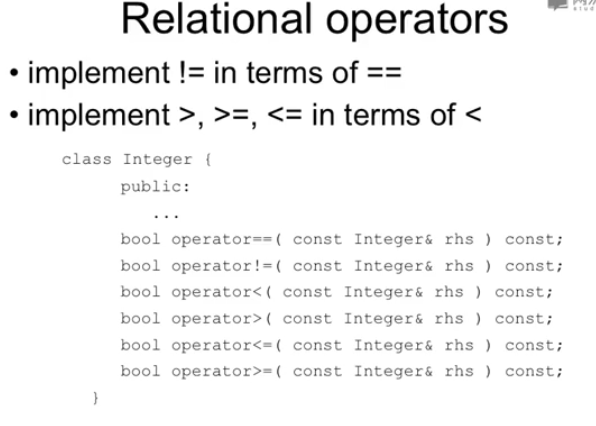


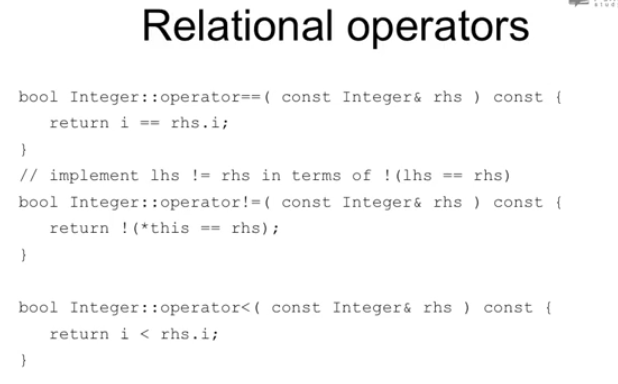
**后缀自增运算符有一个 整型参数**！

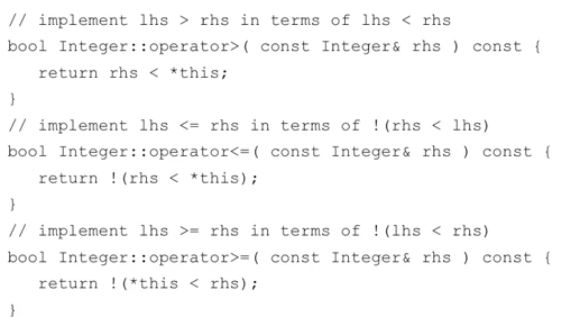


对于自增运算符而言，其返回值类型依具体情况而定。

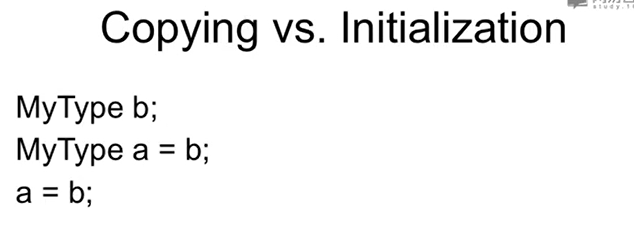


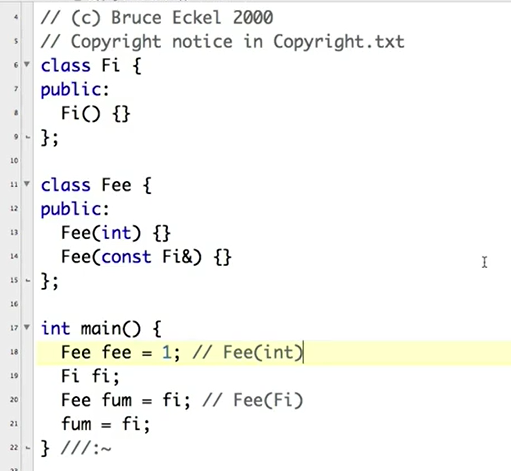


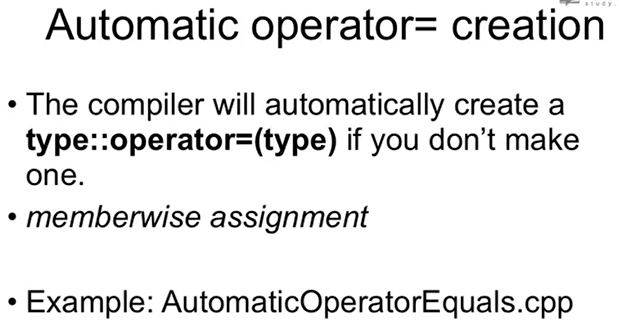


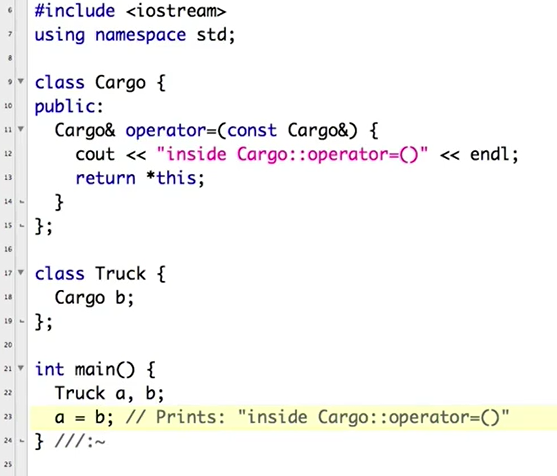


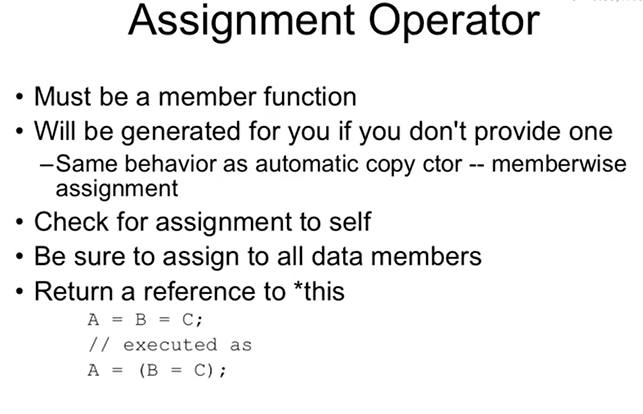
# 32 赋值

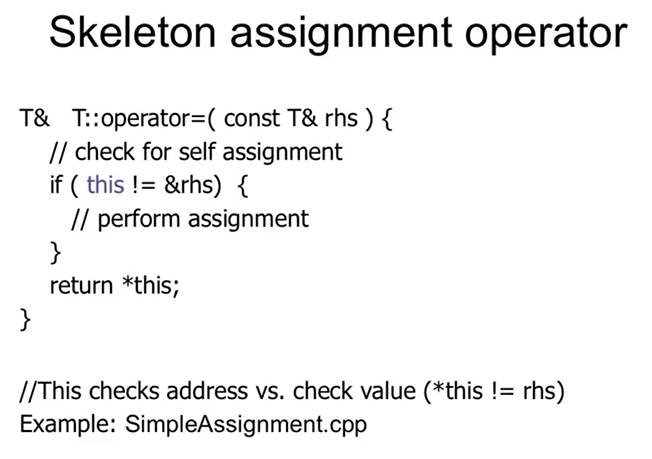




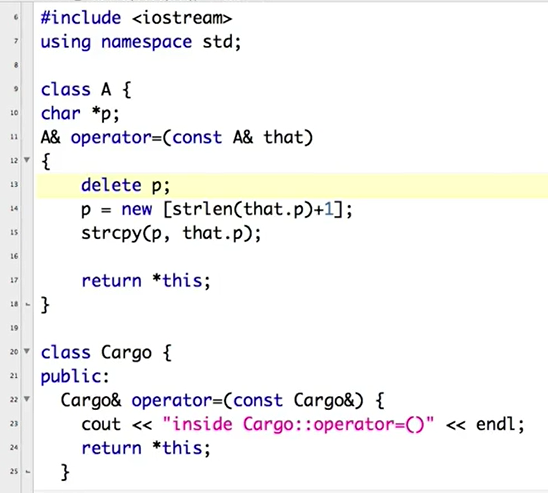


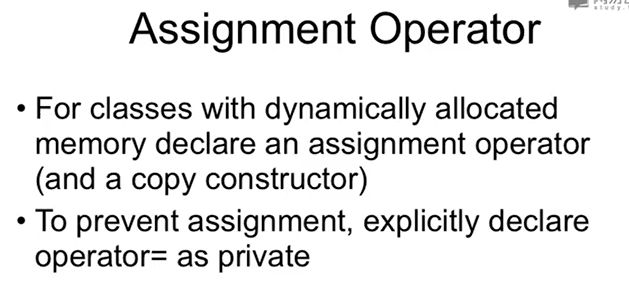




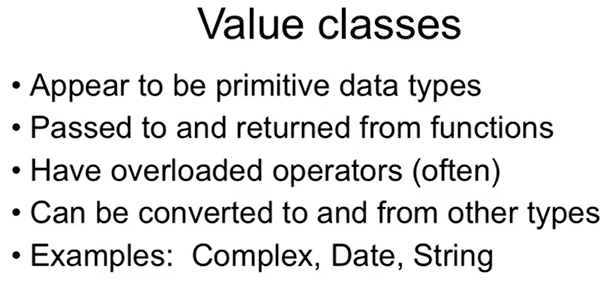


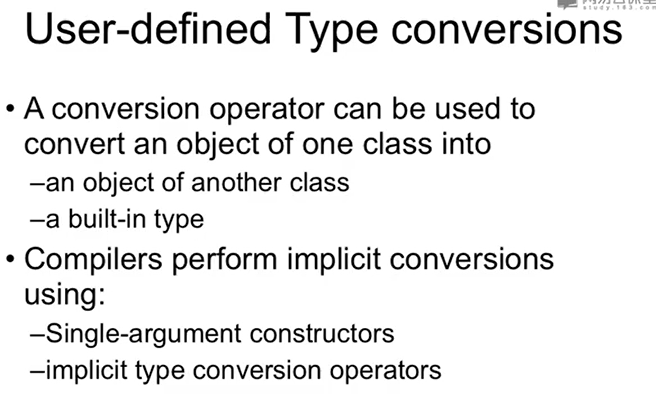
**为了要做this != &rhs 的检查？**

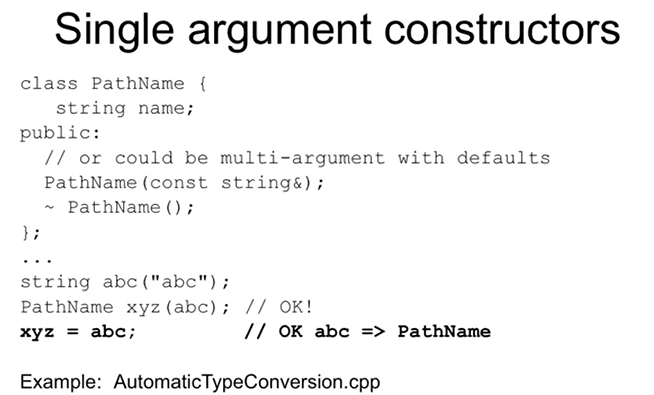




# 33 类型转换







首先将abc使用构造函数做成一个PathName

再使用default operator赋值，

