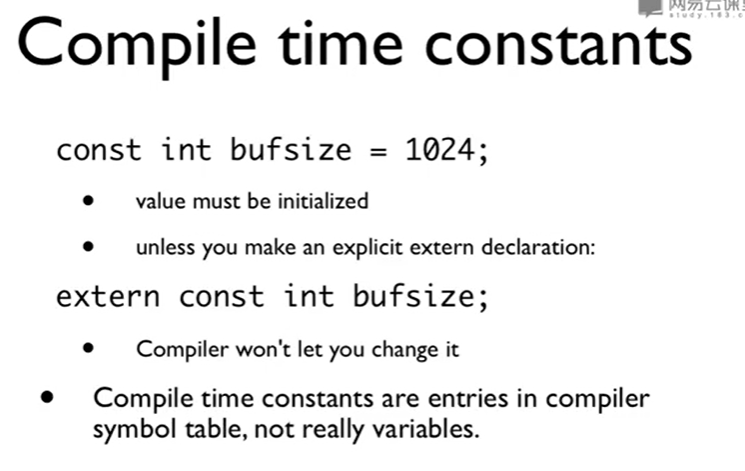
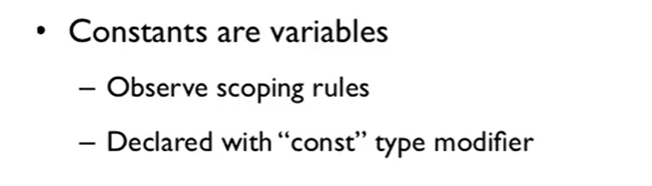
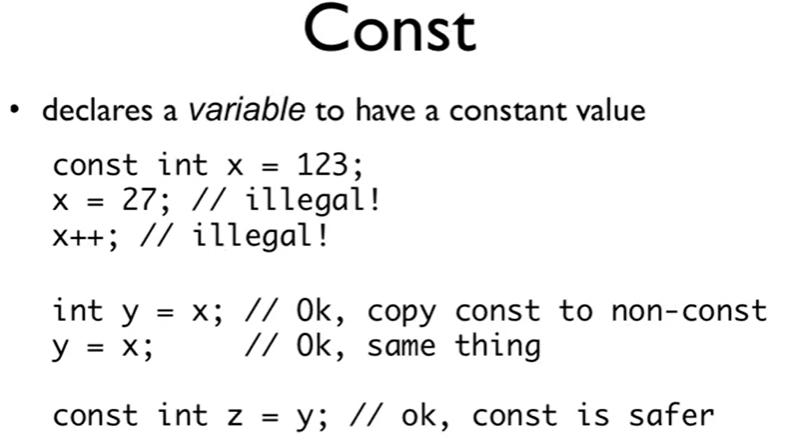
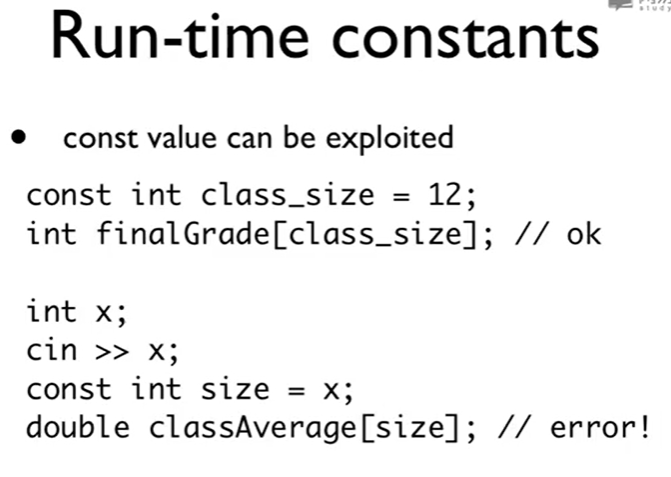
* 常变量/指针
* 编译时常量
* 常返回值？
* 函数调用时，常参地址
* 常成员函数/常数据成员
* 常对象



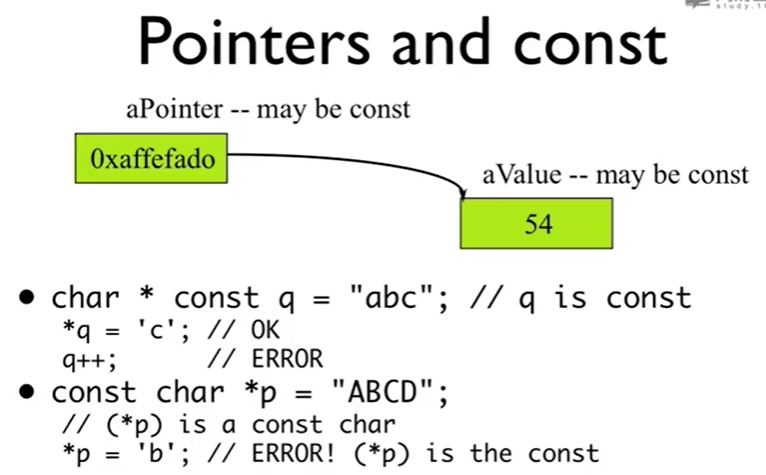
编译时常量

const int bufsize = 1024;

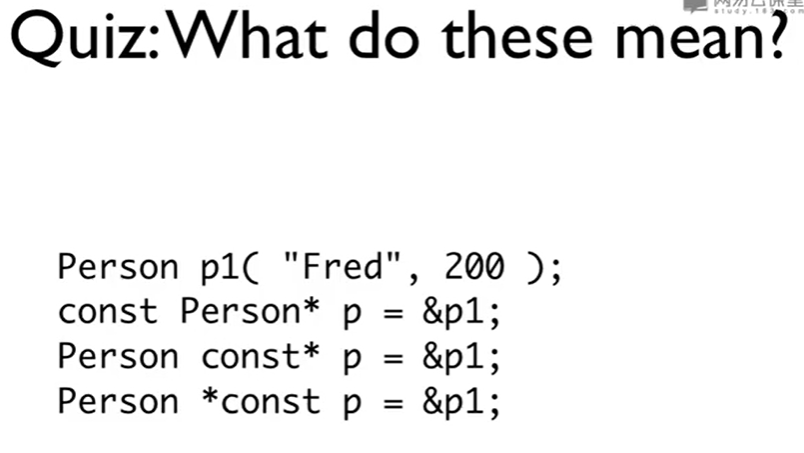
* 常对象的值必须被初始化，除非使用extern进行声明。
* 编译器不允许对其值进行修改。
* 编译时常量处于**编译器的符号表中**，并非真正的变量variable.

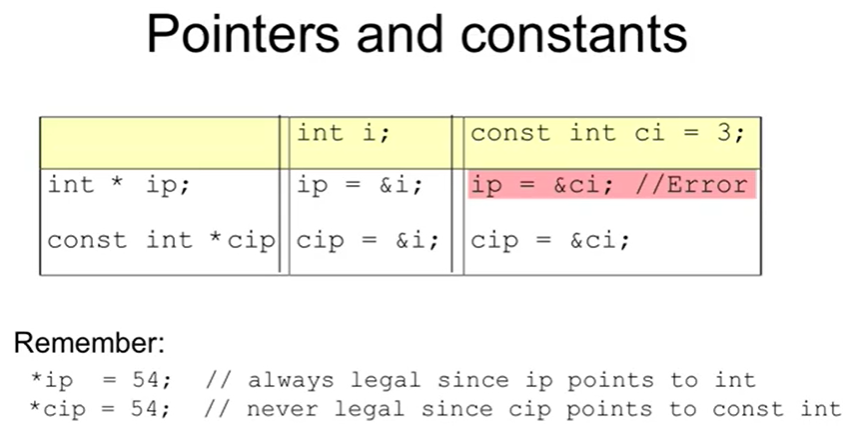


数组大小是在compile time分配？



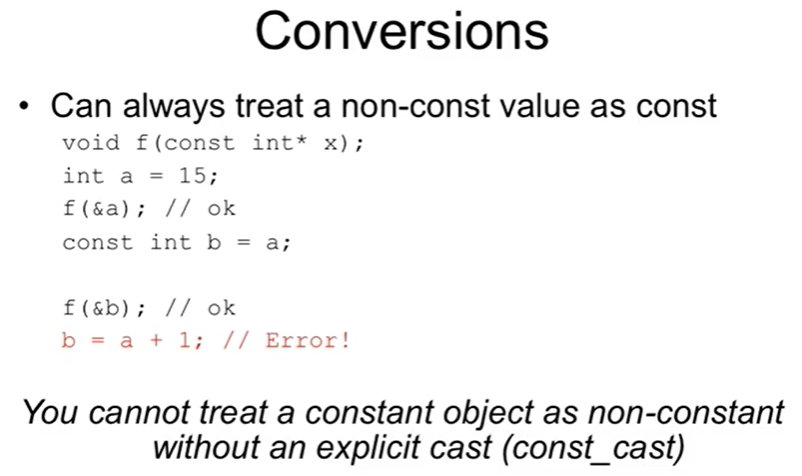
* 指向字符变量的**常指针**
  + **char \* const q**
* **常对象指针**
  + const char \*p

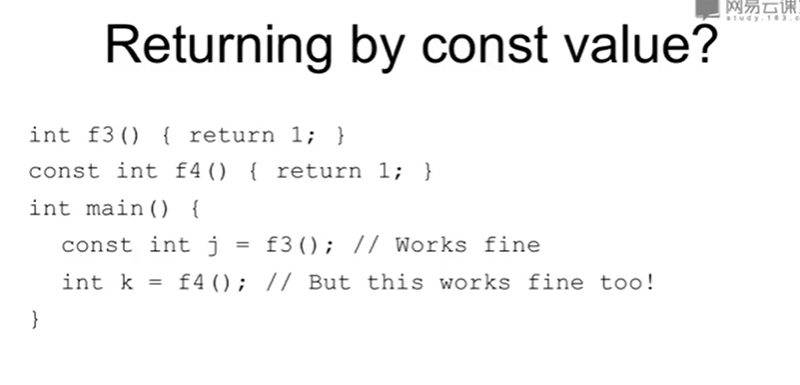


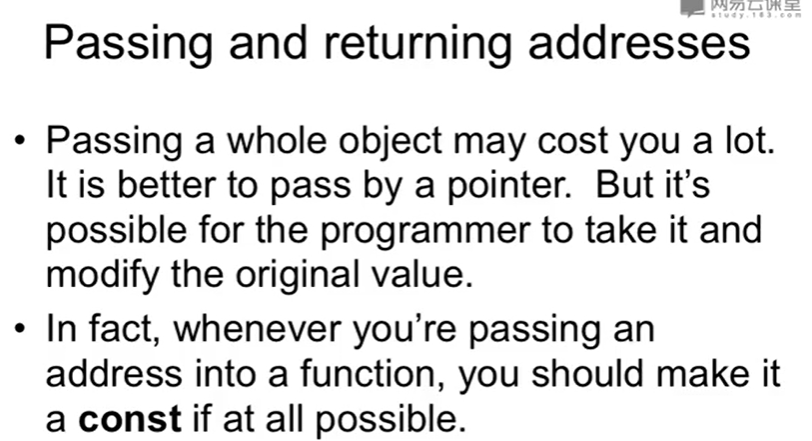


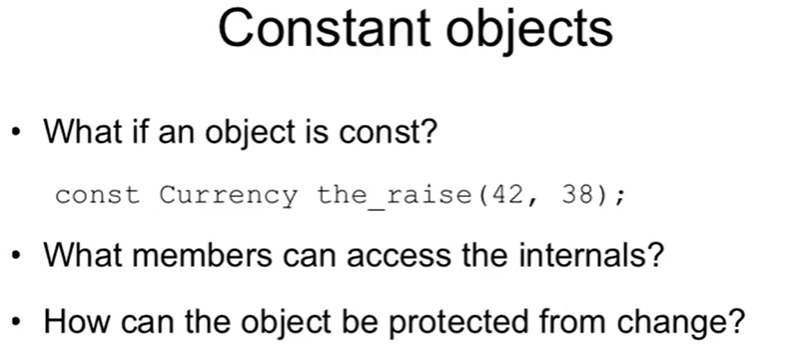
\*ip = 54, 将整型对象赋予整型指针

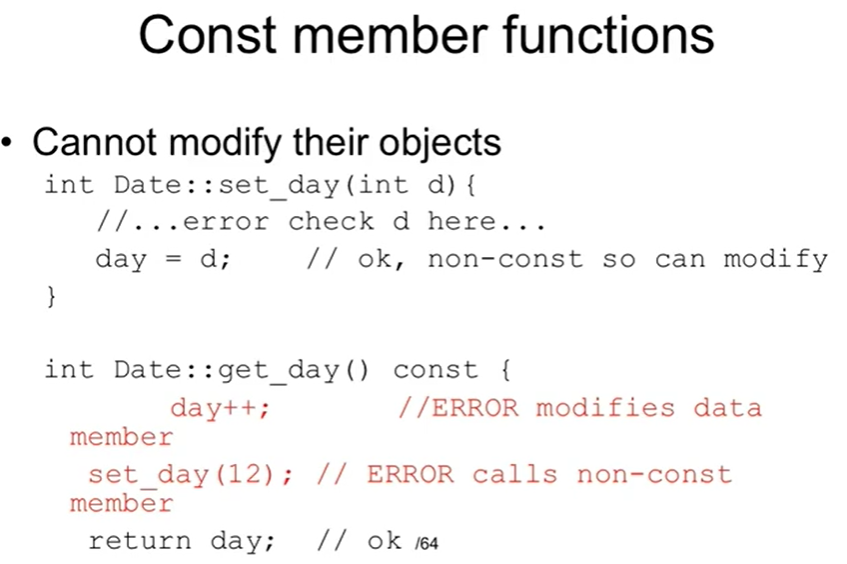
\*cip = 54, 将整型对象赋予常整型对象指针， illegal since cip是指向常整型对象的指针



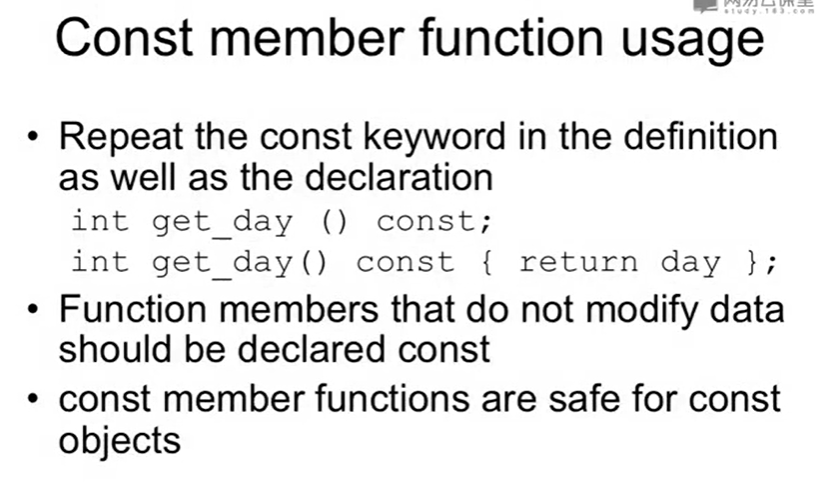








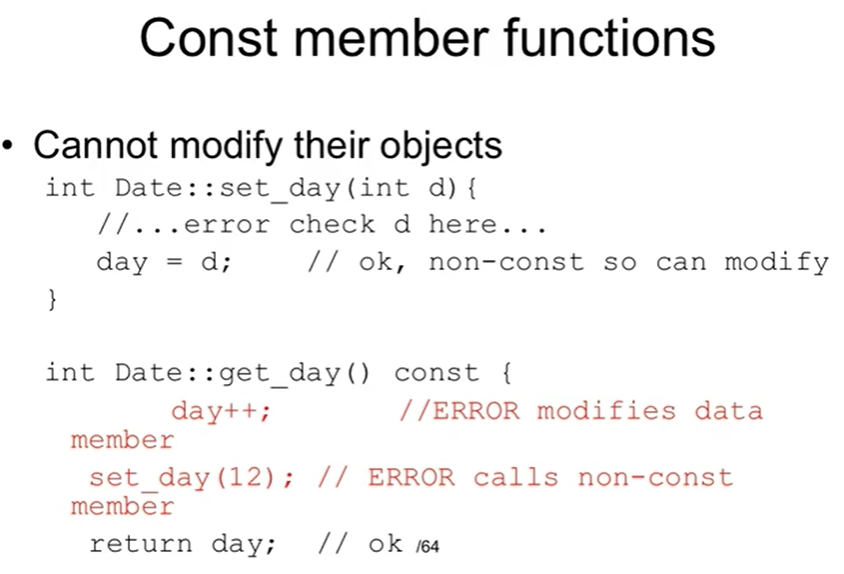
**常成员函数**

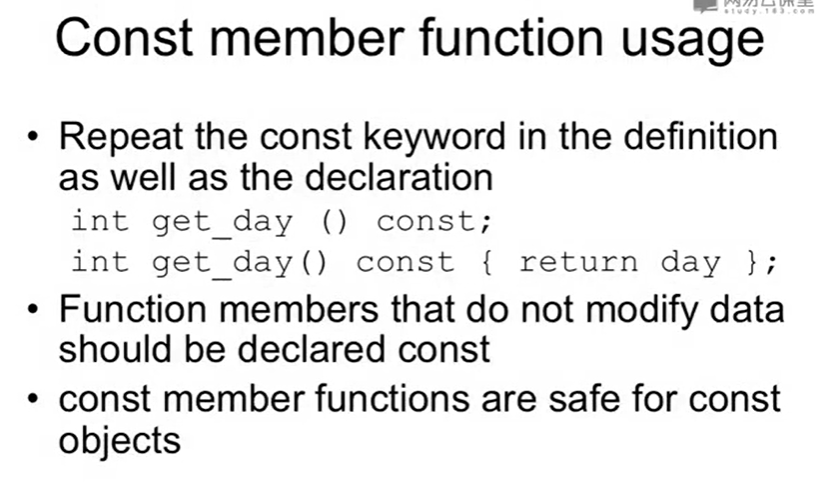


**[将const写在函数名括号后面，表明this是const**？]

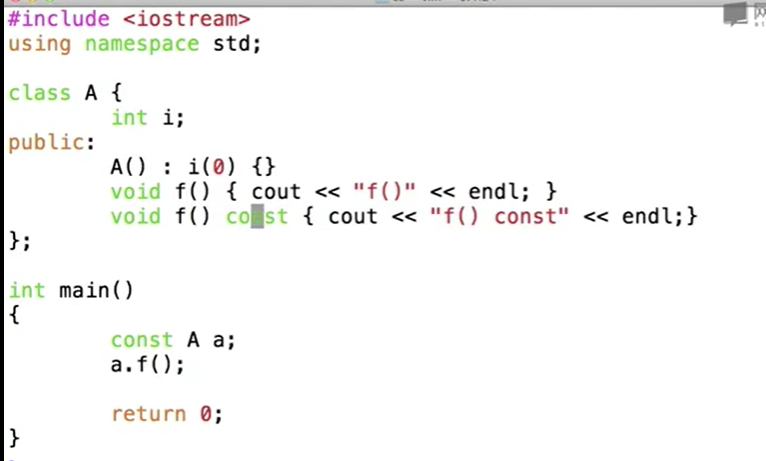
* 在**声明和定义处均需要在参数后跟上const关键字**
* 不修改数据的函数成员应该被声明为**const**
* **常成员函数对常对象比较安全**

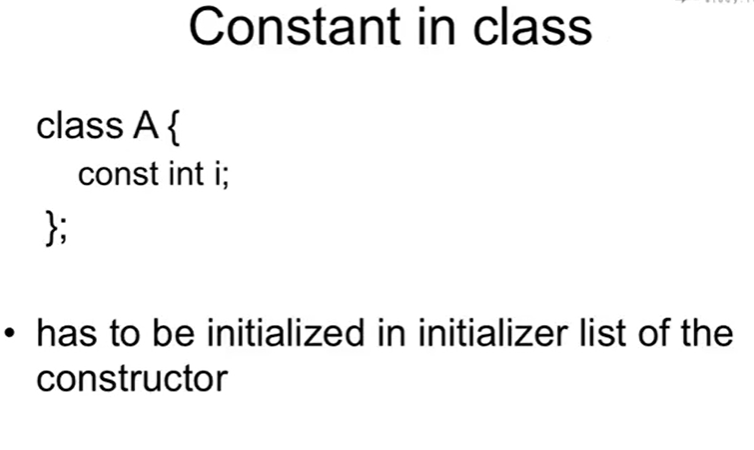
# # 20 不可修改对象



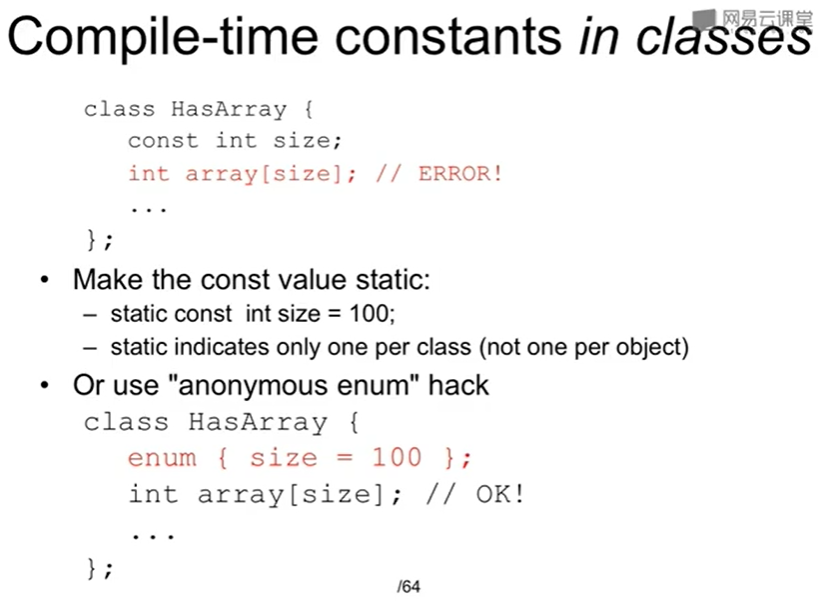


* 常成员函数不能修改**数据成员**和调用**非常成员函数**





* 类中的**const需要在初始化列表中进行初始化**
  + **const常对象需要在定义时进行初始化**



* 类中编译时常量

static

enum { size=100 }