

C 箴言：让=返回一个\*this的引用 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_C\\_\\_\\_E7\\_AE\\_B4\\_E8\\_A8\\_80\\_EF\\_c97\\_133681.htm](https://www.100test.com/kao_ti2020/133/2021_2022_C___E7_AE_B4_E8_A8_80_EF_c97_133681.htm) 关于赋值的一件有意思的事情是你可以把它们穿成一串。 `int x, y, z. x = y = z = 15. // chain of assignments` 另一件有意思的事情是赋值是右结合的，所以，上面的赋值串可以解析成这样：`x = (y = (z = 15))`。这里，15赋给z，然后将这个赋值的结果（最新的z）赋给y，然后将这个赋值的结果（最新的y）赋给x。这里实现的方法就是让赋值运算符返回一个左侧参数的引用，而且这就是当你为你的类实现赋值运算符时应该遵守的约定：`class Widget { public: ... Widget&. rhs) // return type is a reference to { // the current class ... return *this. // return the left-hand object } ... }`。这个约定适用于所有的赋值运算符，而不仅仅是上面那样的标准形式。因此：`class Widget { public: ... Widget&. rhs) // the convention applies to { // =, -=, *=, etc. ... return *this. } Widget&. operator=(int rhs) // it applies even if the { // operator ' s parameter type ... // is unconventional return *this. } ... }`。这仅仅是一个约定，代码并不会按照这个意愿编译。无论如何，这个约定被所有的内建类型和标准库中（或者即将进入标准库）的类型（例如，string，vector，complex，tr1::shared\_ptr等）所遵守。除非你有好的理由作些不同的事情，否则，不要破坏它。

Things to Remember 让赋值运算符返回一个\*this的引用。

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](http://www.100test.com)