

二级考试C 辅导:有关缺省变量值的限制计算机二级考试 PDF  
转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/558/2021\\_2022\\_\\_E4\\_BA\\_8C\\_E7\\_BA\\_A7\\_E8\\_80\\_83\\_E8\\_c97\\_558188.htm](https://www.100test.com/kao_ti2020/558/2021_2022__E4_BA_8C_E7_BA_A7_E8_80_83_E8_c97_558188.htm) 在准标准C中，有关缺省变量值的限制非常模糊。基于此，很多编译器允许开发人员将缺省变量值包含在函数声明，指向函数的指针和引用，成员函数的指针，以及typedef声明中。请看一下以下的程序：  

```
struct A { void func(int x=5) { } }. void g(int n=12) { } // 根据C标准，不能在以下声明中使用缺省变量值。 void (*pf)(inti=120). void (A::*pmf)(int j=50). typedef void (*PF)(inti=100). // 函数的引用 typedef void (amp.A::func. A a. // 这些调用使用了哪些缺省值？ pf(). pf2(). (a.*pmf)(). } A::func()和g()具有缺省变量值，这是合理的。然而，指针pmf，pf以及typedefPF也定义了缺省的变量值。根据C标准，这是不规范的。这一代码的实际使用中，其中的一个问题是这些声明中提供的缺省值与A::func()和g()函数提供的值不一致。也就是说，很多编译器将这些代码作为非标准的扩展。当调用g()函数时，我的编译将120作为pf的缺省值；然而，对于pf2，它使用100作为它的缺省值。作为一种规则，应该避免使用指向函数的指针，成员函数的指针，以及typedef命名的缺省变量值。即使你的编译器接受了它们，在更高版本中它也可能不被接受。而且，这些代码也会降低程序的灵巧性，也会给那些无法判别哪些编译器接收何种缺省变量的开发人员带来误导。在使用这些缺省变量值的合法代码中，这里的建议是添加一些必要的注释，以说明需要哪些缺省变量值。2009年上半年全国计算机等级考试参考答案请进入计算机考试论坛
```

2009年上半年全国计算机等级考试报名信息汇总 2009年NCRE  
考试有新变化 2009年全国计算机等级考试大纲 2009年上半年  
全国计算机二级考试试题及答案 2009年上半年全国计算机等  
级考试试题答案汇总 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)