从VC 到GCC移植：谈两者语法差异 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/271/2021_2022__E4_BB_8EVC___E5_88_B0_c97_271875.htm 类型引用template class Foo { typedef T::SomeType SomeType. }.这段代码在VC 中一点问题也没有，但是GCC并不允许，因为它不知道T::SomeType是什么。你需要改为：template class Foo { typedef typename T::SomeType SomeType. }.通过typename T::SomeType告诉GCC，SomeType是一个类型名，而不是其他东西。当然，这种情况不只是出现在typedef中。例如：template void visit(const Container&amp. cont) { for (Container::const_iterator it = cont.begin(). it != cont.end(). it) ... }这里的Container::const_iterator同样需要改为typename Container::const_iterator。基类成员引用template class Foo : public Base { public: void foo() { base_func(). m_base_member = 0. } }.这段代码在VC 中同样没有问题，但是GCC中不能通过。因为GCC并不知道base_func，m_base_member是什么。对于这个问题，你可以有两种改法：改法1：加上域作用符Base::template class Foo : public Base { public: void foo() { Base::base_func(). Base::m_base_member = 0. } }. 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