

转：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_ti2020/286/2021_2022__E8_BD_AC_EF_BC_9A_E4_BB_8EV_c67_286796.htm 类型引用 以下是引用片段：

```
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：
```

```
以下是引用片段： template void
```

```
visit(const Container& 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 以下是引用片段：

```
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:: 以下是引用片段：

```
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