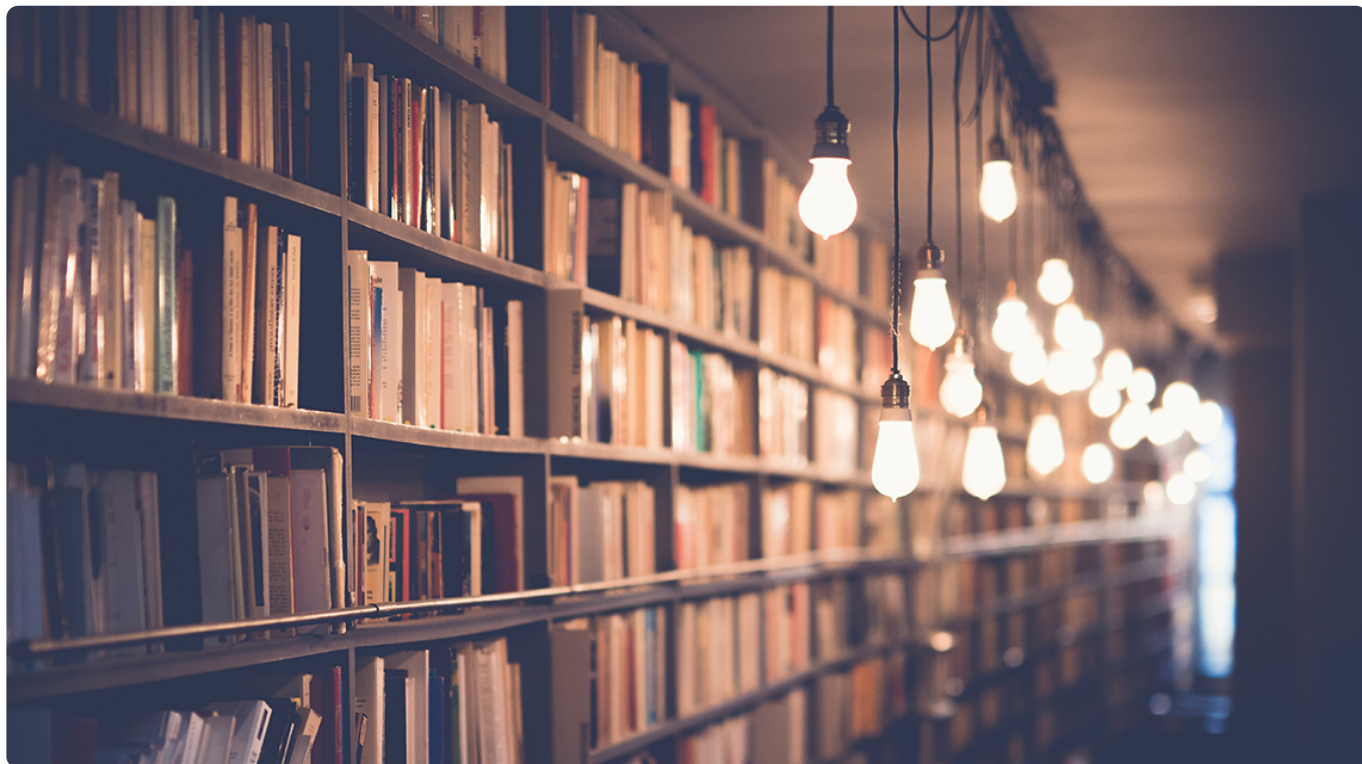


课前必读 | 有关术语发音及环境要求

2019-11-25 吴咏炜

现代C++实战30讲

[进入课程 >](#)



讲述：吴咏炜

时长 04:46 大小 3.28M



你好，我是吴咏炜。

这一讲是课前必读，主要说明术语发音和环境要求的问题，希望能帮助你更好地学习和实践本专栏的内容。

术语发音列表

C++ 里有些术语不是标准的英文单词。在本专栏中使用下面的发音表：

术语	发音
asm	/ˈæz(ə)m/
char	/tʃɑ:(r)/
constexpr	/ˈkɒnstˌekspə(r)/
cout	/ˌsiːˈaʊt/
decltype	/ˈdek(ə)ltaɪp/
decltype	/ˈdek(ə)lvæl/
-ptr	/pɒtə(r)/ (如 nullptr /ˈnʌlˌpɒtə(r)/)
RAII	/ˌɑːreɪaɪˈaɪ/
SFINAE	/ˌesfiˈneɪ/
std	/stʊd/
wchar_t	/ˌdʌbljuːtʃɑ:(r)ˈtiː/
*	star /stɑ:(r)/
&	ref /ref/
->	arrow /ˈæərəʊ/

注意这些发音并没有标准化，列出来只是为了避免发生误解。你不一定必须按照我的读法来。

环境要求

在专栏中，我大部分时间都会使用不超过 C++17 标准的语言特性。而要编译示例代码，需要以下所列的至少一种编译器：

GCC 7 或更新版本

Clang 5 或更新版本

Visual Studio 2017 15.9 (MSVC 19.16) 或更新版本

以上三种编译器应当都能够工作——但我无法保证。如果遇到问题的话，请升级到跟我的测试环境相同的版本：

GCC 8.3

Clang 9.0

Visual Studio 2019 16.3 (MSVC 19.23)

如果你对跨平台性不那么讲究的话，推荐使用你的操作系统里最常用的编译器。也就是：

用 Windows 的话，使用 MSVC

用 Linux 的话，使用 GCC

用 macOS 的话，使用 Clang（但如果操作系统比 Mojave 老的话，苹果提供的开发工具不能完整支持 C++17；你需要升级操作系统，或者使用非苹果的编译器——如 Homebrew 提供的 gcc 和 llvm [\[1\]](#)）

对在 Windows 上使用 GCC 的开发者，我要特别提醒一句：要获得最全面的功能，你应当使用 MinGW-w64 的 POSIX 线程版本，这样才能完整使用 C++ 标准里的功能。当前版本可从参考资料 [\[2\]](#) 的链接下载。

使用稳定发布版（如 CentOS）的 Linux 用户也需要检查一下，你的 GCC 版本有可能比较老。如果早于 GCC 7 的话，建议你安装一个新版本的 GCC（不需要覆盖系统的 GCC）。比如，对于 CentOS 7，系统安装的 GCC 版本是 4.8，太老，你可以通过安装 `centos-release-scl` 和 `devtoolset-7-gcc-c++` 两个包来获得 GCC 7；随后，可以使用命令 `scl enable devtoolset-7 bash` 或 `./opt/rh/devtoolset-7/enable` 来启用 GCC 7。

稍需注意的是，最后在讲到 C++20 新特性时，某些实验功能可能会要求某个特定的编译器。这种情况下，你可能就需要安装、使用非默认的编译器了。不过，只有少数几讲需要这么做，不用担心。

由于专栏涉及到的都是较短的代码，我不会提供工程文件。建议你熟悉编译器的命令行，来快速编译代码。使用 GCC 的话，推荐使用下面的命令行：

```
g++ -std=c++17 -W -Wall -Wfatal-errors 文件名
```

Clang 的话也比较类似：

```
clang++ -std=c++17 -W -Wall -Wfatal-errors 文件名
```

MSVC 的命令行风格有点不同，一般需要下面这样子：

```
cl /std:c++17 /EHsc /W3 文件名
```

另外，即使不用较新的 C++ 特性，你也一定要用比较新的编译器。单单是输出错误信息的友好程度，老版本和新版本就是没法比的。

以 GCC 为例，老版本输出错误信息是单色的，在碰到有模板的代码时，错误信息动辄几百行，以致那时有人专门开发了软件来让错误信息更可读 [3]。幸运的是，我们今天不再需要这类软件了，编译器的输出比之前友好得多，GCC 和 Clang 还会使用颜色来展示问题的重点。下面这张图，就是 GCC 9.2 的输出。

```
hello_world.cpp: In function 'int main()':
hello_world.cpp:5:5: error: 'cout' was not declared in this scope; did you mean
'std::cout'?
   5 |         cout << "Hello World!" << endl;
     |         ^~~~
     |         std::cout
compilation terminated due to -Wfatal-errors.
```

明确好以上内容，我们就要正式开始了，你准备好了吗？

参考资料

[1] Homebrew. <https://brew.sh/>

[2] MinGW-w64 GCC-8.1.0. <https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/8.1.0/threads-posix/dwarf/>

[3] Leor Zolman, "STLFilt: An STL error message decryptor for C++" .

<https://www.bdsoft.com/tools/stlfilt.html>


点击查看 

打卡学习 C++ 拒绝从入门到放弃



PC 端用户扫码参与



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 [开篇词 | C++这么难，为什么我们还要用C++？](#)

下一篇 [01 | 堆、栈、RAII：C++里该如何管理资源？](#)

精选留言

 写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。