

结束语 | 路远，未有穷期

2020-06-30 罗剑锋

罗剑锋的C++实战笔记

[进入课程 >](#)



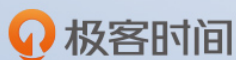
罗剑锋

奇虎360技术专家，Nginx/OpenResty开源项目贡献者

你好，我是罗剑锋。

C++ 是一门高度复杂自洽的编程语言，我们要有“战略上藐视，战术上重视”的觉悟，不贪大，不求全，踏踏实实地去学习、应用，才能做到“日日精进”。

愿我们在 C++ 的旅途上永远开心，不留半点遗憾！



讲述：Chrono

时长 08:02 大小 7.37M



你好，我是 Chrono。

两个月的时间过得好快啊，一眨眼的功夫，就到了和你说“再见”的时候了。

和之前的《透视 HTTP 协议》一样，我首先要感谢的，还是你——感谢对我和这门课的支持，正是你持续的支持和鼓励，才让我有了创作的激情和动力，Thanks for your support!

同时，我也要祝贺你，祝贺你凭着力量、智慧和勇气（《The Legend of Zelda》的真三角），突破了 C++ 的“重重迷雾”，成功地掌握了 C++ 的核心知识和应用技能，相信这将会是你今后工作履历中的一笔“浓墨重彩”。



现在再回头看各节课程，我觉得基本上算是达成了当初预计的目标：贴近实际的开发经验分享，没有剑走偏锋的“葵花宝典”，而是踏踏实实的“太祖长拳”。

但是我觉得这门课程的交付还是有一点点遗憾的。因为 C++ 的内容实在是太丰富了，在课程里，我们只能挑出“必修”的重点和要点来讲，还有很多外围的点没办法展开细讲，还有很多小点来不及说出来，真是舍不得和你说再见。

所以，我想在这最后一讲里，我还能和你交代点啥，我觉得最实际的就是和你聊聊结课后，你该怎么继续学习 C++。

课程结束后，你该怎么学 C++？

要回答这个问题，我们再来看看这门课的主角 C++ 吧。

从 1979 年贝尔实验室发明“C with Classes”，到 1983 年正式命名为“C++”，再到现在，C++ 已经走过了四十多年，可以说是处于“不惑之年”了。在如今的编程语言界，和 Java (1995)、Go (2009)、Rust (2010) 等比起来，C++ 真算得上是“老前辈”。

这么多年来，C++ 一直没有停止发展的脚步（不过确实有点缓慢），不仅修补了曾经的缺陷，还增加了越来越多的新特性。不可否认，C++ 虽然变得越来越复杂和庞大，但也正在一步步趋近“完美”，你可以批评它，但绝不能无视它。

个人认为，**C++ 最大的优点是与 C 兼容，最大的缺点也是与 C 兼容。**

一方面，它是 C 之外唯一成熟可靠的系统级编程语言（目前 Rust 还没有达到可以和 C++ “叫板”的程度），大部分用 C 的地方都能用 C++ 替代，这就让它拥有了广阔的应用天地。而面向对象、泛型等编程范式，又比 C 能够更好地组织代码，提高抽象层次，管理复杂的软件项目。比如前段时间刚成功发射的“龙”飞船，它的应用软件就是用 C++ 开发的。

但另一方面，为了保持与 C 兼容，C++ 的改革和发展也被“束缚了手脚”，做出任何新设计时，都要考虑是否会对 C 代码造成潜在的破坏。这就使得很多 C++ 新特性要么是“一拖再拖”，要么是“半成品”，要么是“古里古怪”，最后导致 C++ 变得有些不伦不类，丢掉了编程语言本应该的简洁、纯粹。

如果你看看新出的 C++17、C++20，这两方面就表现得特别明显（比如结构化绑定、模块、协程、模板 lambda）。也许，这就是 C++ 追求自由和性能的代价吧。

这次的课程里，我们主要用的是 C++11/14，**那么面对后续的新标准、新特性，在没有“入职导师”的情况下，你要再怎么学呢？**

我觉得，对于 C++ 这样复杂的编程语言，你要把握一个基本原则：**不要当“语言律师”**（language lawyer）。也就是说，不要像孔乙己那样，沉迷于“茴”字有多少种写法，又或者是“抖机灵”式的代码，而是要注重实践、实用。

因为 C++ 的编程范式太多，“摊子”实在是铺得太大，它为了避免各种特性可能导致的歧义和冲突，就会有许许多多细致到“令人发指”的规定，我们在学习的时候，一不小心就会钻进细节里出不来了。

这样的例子有很多，比如说 ADL、引用折叠、可变参数模板、“++”的前置和后置用法、模板声明里的 typename 和 class、初始化列表与构造函数、重载函数里的默认参数.....

弄懂这些位于“犄角旮旯”里的特性（无贬义），需要花费我们很多的脑力，但在我们一般的开发过程中，通常很少会触及这些点，或者说是会尽力避免，它们通常只是对编译器有意义，所以在这些“细枝末节”上下功夫就不是很值了，说白了，就是性价比太低。

我个人认为，在掌握了专栏里 C++11/14 知识的基础上，如果再面对一个 C++ 新的语言特性，你不能够在五分钟（或者再略长一点）的时间里理解它的含义和作用，就说明它里面的“坑”很深。

你应当采用“迂回战术”，暂时放弃，不要细究，把精力集中在现有知识的消化和理解上，练好“基本功”，等你以后真正不得不用它的时候，通过实践再来学习会更好。

这也是我自己多年实践的经验，希望对你有用。

结课赠言

结束语的最后，我还是要来一个“首尾呼应”，回顾一下开篇词里的“初心”。

还记得开课时的那三句编程格言吗？

任何人都能写出机器能看懂的代码，但只有优秀的程序员才能写出人能看懂的代码。

有两种写程序的方式：一种是把代码写得非常复杂，以至于“看不出明显的错误”；另一种是把代码写得非常简单，以至于“明显看不出错误”。

“把正确的代码改快速”，要比“把快速的代码改正确”，容易得太多。

这两个月的学习下来，结合 C++ 的各种编程范式、特性和应用经验，你是否对它们又有了一些新的认识呢？

我说说我对这三句格言的理解吧，也许能和你产生共鸣：

写代码，是为了给人看，而不是给机器（编译器、CPU）看，也就是 human readable；

代码简单、易理解最重要，长而复杂的函数、类是不受欢迎的，要经常做 Code Clean；

功能实现优先，性能优化次之，在没有学会走之前，不要想着跑，也就是 Do the right thing。

课程即将结束，在漫长的学习道路上，它的作用就好像是一个小小的驿站，让你适时停下来休息，补充点食物和清水，为下一次冒险做好准备。希望在你将来的回忆里，还会记得有这么一个给你安心和舒适的地方。

这里我还给你准备了一份毕业问卷，题目不多，希望你能抽出几分钟时间填写一下。我非常愿意听听你对这个课程的反馈和建议，你可以在问卷中畅所欲言。

＝ 在 7 月 7 日前提交问卷，将有机会 ＝

得

极简数码收纳包



或得

极客时间课程阅码

价值 **¥99**



好了，临别之际，在你上马、踏上新的征途之前，我再送你一句“老话”吧：

一个人写出一个好程序不难，难的是一辈子只写好程序，不写坏程序。

路远，未有穷期，期待我们的下次再会！

618 课程特惠

618 好课 5 折起

优惠口令立减 ¥15

618gogogo



© 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 期末测试 | 这些C++核心知识，你都掌握了吗？

下一篇 轻松话题（一） | 4本值得一读再读的经典好书

精选留言 (11)

写留言

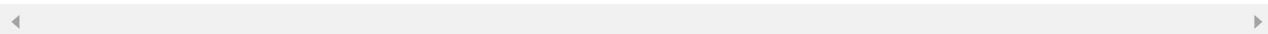


种花家

2020-06-30

正在学习c++的小白，每次重复听罗老师讲课，就对c++理解和认识有重了一分

作者回复: 课下再多结合GitHub源码，自己把代码多敲敲，多练习，收获就会更大。



1

1

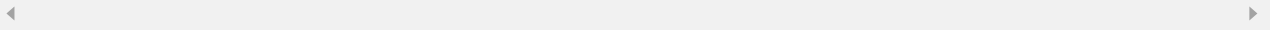


EncodedStar

2020-06-30

有机会，罗老师给我们发一些加餐呀，感谢罗老师的这节c++，收获颇多

作者回复: 加餐需要考虑选题, 暂时没想好有哪些内容, 欢迎大家留言出主意。



3

1



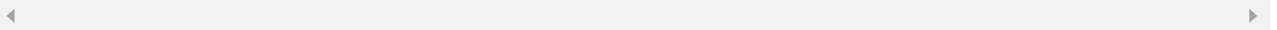
心在飞

2020-06-30

感谢罗老师一路相伴, 后会有期。

展开

作者回复: 应该还会有机会。



1

1



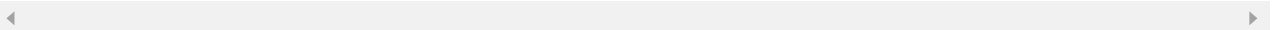
kenan

2020-06-30

罗老师意犹未尽啊! 来个升级版, 期待不久回归。

展开

作者回复: 有问题可以随时回来一起讨论。



1

1

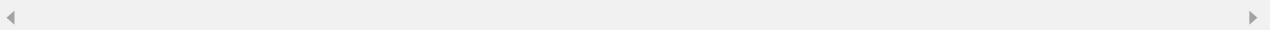


DCRS

2020-06-30

前面的课程, 每次重新听都有新的收获, 感谢罗老师的课程。

作者回复: 不客气, 欢迎常回来看看。



1

1



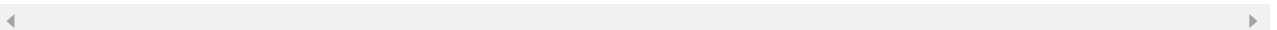
黄骏

2020-06-30

罗老师还在写代码么?

展开

作者回复: 一直在写啊, 不愿意做管理工作, 还是代码最实在。





无为而立

2020-06-30

感谢罗老师的宝贵经验，看老师写的东西就像聆听一位长者在讲述他的经历和故事

作者回复: 言重了，我还年轻啊，笑。



2



Weining Cao

2020-06-30

感谢老师的总结，虽然我觉得这门课结束的有点快。。。理论和实践总是会有一些距离。前面看老师的理论阐述感觉很好理解，可是一到书店的例子看到那么多陌生的类库和用法又有些发懵，也许是因为例子涵盖了网络编程的不少知识吧，所以我又乖乖地去定了老师的http课程：)

展开 ∨

作者回复: 有C++方面的疑问可以随时留言，大家一起交流。



有学识的兔子

2020-06-30

根据老师推荐的书，在网上淘了本《C++程序设计及演化》从struct C 到 看到了虚函数的引入，还是有不少收获。

展开 ∨

作者回复: 这本书是很好的，有点哲学的味道。



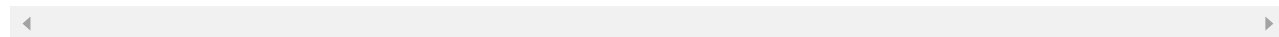
lckfa李钊

2020-06-30

课程学完了，确实有些意犹未尽，最大的感触是，我居然对lua感兴趣了，最近正在新学《透视HTTP》😁，说到底任何编程语言都是工具，我也从不觉得C++是最好的语言，结语里的建议适用于所有开发人员...现在在Http的课程下留言，老师是否看得到呢，毕竟已经结课了

展开 ▾

作者回复: 我每天都会抽出一点时间来看留言的,基本上是有问必答,和其他课程比起来,我的留言回复率应该是最高的吧,笑。



Luca

2020-06-30

再会!

展开 ▾

作者回复: see you next mession.

