

C++ For Beginners

Lesson 0: Overview

0.1 认识 C++

“If I have been able to see further, it was only because I stood on the shoulders of giants.” (如果说我看得比别人更远些, 那是因为我站在巨人的肩膀上。)

--Isaac Newton (艾萨克·牛顿)

你可能时常看到过诸如“C 语言”、“C++”等字眼, 并且常常感到迷惑, 所谓“C/C++ 语言”究竟是何方神圣。事实上, C 语言和 C++ 语言只是两门不同的编程语言, 其中 C 作为最古老、最初级的编程语言为计算机的每一个组件留下了它的足迹。C++ 是 C 的拓展, 但 C 并非 C++ 的子集。C++ 作为一门独立的语言早已与 C 语言分道扬镳, 可以说 C++ 沿用了 C 语言的经典语法, 然后在此基础上加入了大量 (远远超过 C 语言原本特性数量) 新特性。

C++ 的设计理念是 “**What you don’ t use, you don’ t pay for.**” (你不会为你不使用的特性付出代价。), 在这个理念的指导下, C++ 将大量其他现代语言需要在运行时处理的问题使用各种 Tricks 在编译时进行静态分析, 不会在运行过程中引入无用的特性。

C++ 并非一门年轻的语言, 相反的, 自它诞生至今已有三十余年, 在这漫长 (相对于简短的计算机史来说) 的三十年中, C++ 引入了大量的现代化特性, 使得 C++ 被分裂成所谓 “Classic C++ ” 与 “Modern C++”; 借助 Modern C++, 你可以以极低的代价重现其他编程语言中的几乎全部优秀特性。

C++ 已经被历史证明他的高效与优秀, 这门历史悠久的语言在今天仍在焕发生机。这门课将在前几章围绕 Classic C++ 展开描述, 为的是使学生能够有一个对计算机体系结构的全面认识; 后几张基于 Modern C++ 展开描述, 能够使学生的思维得到开拓。

0.2 课程目标

本课程设立的目的是给零基础的学生讲解计算机的基础知识, 全课程使用 C++ 语言, 每节课程时长一小时到两小时不等, 共九节课程, 均附有讲义和课后作业。每节课程都在作者的 B 站直播间讲解, 直播录像将上传到作者的 B 站账号下, 所有讲义可以在作者的个人网站下载。

现课程规划如下:

1. 第一节课将会通过一个简单的 C++ 程序示例展开对计算机体系的讲解。
2. 第二节课将介绍 C++ 的基本概念与语法 (变量、常量、类型、表达式、语句等)。
3. 第三节课将介绍 C++ 的函数和控制流概念。(函数、条件、循环等)。
4. 第四节课将介绍 C++ 的面向对象编程基础。(类、继承等)。

5. 第五节课将介绍 C++ 的泛型编程。(模板、类型推导)。
6. 第六节课将介绍 C++ 的一些常用标准库 API (STL 等)。

每门课程都分为基础课程和进阶课程两部分, 建议学生先按顺序学习每一节的基础课程, 并且完全掌握课程后再进行学习。进阶课程主要从原理和行为上剖析 C++ 特性的本质和实现。

0.3 写给学生

看到这里你可能有一个疑问, 为什么我要学习 C++? 事实上, 通过本课程并不能让你掌握什么实际的知识, 因为本课程讲解的只是计算机领域的皮毛, 而笔者水平也不过堪堪入门而已, 课程中难免遗漏及谬误。几十年的高速发展使得计算机领域有了深厚的积淀, 而编程语言不过是一个凿开积淀的工具。如果刨根问底地看待本课程, 那么其中的每门课程都能拆成几本书籍来讲解, 而笔者也不可能完全明白这些能拆成几十本书籍的细节。一个专业的 C++ 工程师很少自称“了解 C++”, 因为这个语言实在是太复杂了, 复杂到可能这个世界上能够完全了解这个语言的人不过几万之数 (也就是 C++ 标准的制定者和 C++ 编译器厂商的工程师), 而在这些人也未必能够全面了解计算机领域的每一个细节。但所幸从事这个行业并不需要全面了解这个领域的所有知识, 通常一个工程师只需要对他从事的细分领域有一个大致的了解和熟练的运用即可胜任。事实上, 很多学艺不精的工程师也可能连这本书介绍的知识都从未了解。如果把计算机领域比作一本书, 那么这本书愈读便愈厚。于笔者而言, 笔者从未想过能够把这本愈读愈厚的书读完, 也许这种事只有人工智能可以胜任。但计算机于笔者而言已经不只是爱好, 应是笔者终其一生的热爱与事业。笔者自认已经在攀登巨人的身体; 现在, 笔者将攀登巨人的经验交付与诸位, 希望我们有朝一日能够登上巨人的肩膀, 俯瞰曾经向往的登峰造极处的景色。

若你在完成课程后还想继续攀登, 那么笔者将在下文向你推荐一些经典著作和笔者阅读的经验, 也许不是最适合你的, 但也是笔者能够提供给你的最好的经验。

0.4 继续向前

- Stephen Prata 的《C++ Primer Plus》 (人民邮电出版社), 无愧于 C++ 经典之名, 是 C++ 初学者的不二之选。
- Randal E. Bryant 与 David R. O'Hallaron 的 *Computer Systems: A Programmer's Perspective* (《深入理解计算机系统》) (机械工业出版社), 书的很多内容早已过时, 但其论述的各种概念和设计都是今天计算机的基础。通过本书可以让你对计算机有一个根本的认识, 但本书难度较大, 需要反复阅读。
- David Vandevoorde 和 Nicolai M. Josuttis 的《C++ Templates》 (人民邮电出版社), 深入剖析了 C++98 的模板特性, 分别由一个程序员的实用主义视角介绍了模板的基本用法和一个学者的严谨视角介绍了模板设计的哲学, 附有大量的 C++ 模板实用例程。
- 欧长坤的 *Modern C++ Tutor* (《现代 C++ 教程 -- 高速上手 C++11/14/17/20》), 篇幅不长却浓缩现代 C++ 的精髓, 可以当作一本参考手册细读。
- 谢丙堃的《现代 C++ 语言核心特性解析》 (人民邮电出版社), 对 Modern C++ 进行成体系的详细介绍, 语言简洁清晰。

- Dmitri Nesteruk 的 *Design Patterns in Modern C++20* (《C++20 设计模式》) (机械工业出版社), 这本书将一些看似简单的现代 C++ 特性运用到实际的问题解决上, 一个优秀的设计模式能够使你规避 Bugs 并且使后期更易于维护, 这点在大型工程上体现尤为明显。