

Lesson00---前言

【本节目标】

- 1. 什么是C++
- 2. C++发展史
- 3. C++的重要性
- 4. 如何学习C++
- 5. 关于本门课程



1. 什么是C++

C语言是结构化和模块化的语言，适合处理较小规模的程序。对于复杂的问题，规模较大的程序，需要高度的抽象和建模时，C语言则不合适。为了解决软件危机，20世纪80年代，计算机界提出了OOP(object oriented programming: 面向对象)思想，支持面向对象的程序设计语言应运而生。

1982年，Bjarne Stroustrup博士在C语言的基础上引入并扩充了面向对象的概念，发明了一种新的程序语言。为了表达该语言与C语言的渊源关系，命名为C++。因此：C++是基于C语言而产生的，它既可以进行C语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行面向对象的程序设计。

2. C++的发展史

1979年，贝尔实验室的本贾尼等人试图分析unix内核的时候，试图将内核模块化，于是在C语言的基础上进行扩展，增加了类的机制，完成了一个可以运行的预处理程序，称之为C with classes。

语言的发展就像是练功打怪升级一样，也是逐步递进，由浅入深的过程。我们先来看下C++的历史版本。











阶段	内容
C with classes	类及派生类、公有和私有成员、类的构造和析构、友元、内联函数、赋值运算符重载等
C++1.0	添加虚函数概念，函数和运算符重载，引用、常量等
C++2.0	更加完善支持面向对象，新增保护成员、多重继承、对象的初始化、抽象类、静态成员以及const成员函数
C++3.0	进一步完善，引入模板，解决多重继承产生的二义性问题和相应构造和析构的处理
C++98	C++标准第一个版本 ，绝大多数编译器都支持，得到了国际标准化组织(ISO)和美国标准化协会认可， 以模板方式重写C++标准库，引入了STL(标准模板库)
C++03	C++标准第二个版本，语言特性无大改变，主要：修订错误、减少多异性
C++05	C++标准委员会发布了一份计数报告(Technical Report, TR1)，正式更名C++0x，即：计划在本世纪第一个10年的某个时间发布
C++11	增加了许多特性，使得C++更像一种新语言，比如：正则表达式、基于范围for循环、auto关键字、新容器、列表初始化、标准线程库等
C++14	对C++11的扩展，主要是修复C++11中漏洞以及改进，比如：泛型的lambda表达式，auto的返回值类型推导，二进制字面常量等
C++17	在C++11上做了一些小幅改进，增加了19个新特性，比如：static_assert()的文本信息可选，Fold表达式用于可变的模板，if和switch语句中的初始化器等
C++20	自C++11以来最大的发行版 ，引入了许多新的特性，比如： 模块(Modules)、协程(Coroutines)、范围(Ranges)、概念(Constraints) 等重大特性，还有对已有特性的更新：比如Lambda支持模板、范围for支持初始化等
C++23	制定ing

C++还在不断的向后发展。但是：现在**公司主流使用还是C++98和C++11**，所有大家不用追求最新，重点将C++98和C++11掌握好，等工作后，随着对C++理解不断加深，有时间可以去琢磨下更新的特性。

3. C++的重要性

3.1 语言的使用广泛度

下图数据来自**TIOBE编程语言社区**2021年12月最新的排行榜，在30多年的发展中，C/C++几乎一致稳居前5。

Dec 2021	Dec 2020	Change	Programming Language		Ratings	Change
1	3	▲		Python	12.90%	+0.69%
2	1	▼		C	11.80%	-4.69%
3	2	▼		Java	10.12%	-2.41%
4	4			C++	7.73%	+0.82%
5	5			C#	6.40%	+2.21%
6	6			Visual Basic	5.40%	+1.48%
7	7			JavaScript	2.30%	-0.06%
8	12	▲		Assembly language	2.25%	+0.91%
9	10	▲		SQL	1.79%	+0.26%
10	13	▲		Swift	1.76%	+0.54%

TIOBE 编程语言社区排行榜是编程语言流行趋势的一个指标，每月更新，这份排行榜排名基于互联网上有经验的程序员、课程和第三方厂商的数量。排名使用著名的搜索引擎（诸如 Google、MSN、Yahoo!、Wikipedia、YouTube 以及 Baidu 等）进行计算。注意：**排名不能说明那个语言好，那个不好，每门编程语言都有适应自己的应用场景。**

3.2 在工作领域

1. 操作系统以及大型系统软件开发

所有操作系统几乎都是C/C++写的，许多大型软件背后几乎都是C++写的，比如：Photoshop、Office、JVM(Java虚拟机)等，究其原因还是性能高，可以直接操控硬件。

2. 服务器端开发

后台开发：主要侧重于业务逻辑的处理，即对于前端请求后端给出对应的响应，现在主流采用java，但内卷化比较严重，大厂可能会有C++后台开发，主要做一些基础组件，中间件、缓存、分布式存储等。服务器端开发比后台开发跟广泛，包含后台开发，一般对实时性要求比较高的，比如游戏服务器、流媒体服务器、网络通讯等都采用C++开发的。

3. 游戏开发

PC平台几乎所有的游戏都是C++写的，比如：魔兽世界、传奇、CS、跑跑卡丁车等，市面上相当多的游戏引擎都是基于C++开发的，比如：Cocos2d、虚幻4、DirectX等。三维游戏领域计算量非常庞大，底层的数学全都是矩阵变换，想要画面精美、内容丰富、游戏实时性搞，这些高难度需求无疑只能选C++语言。比较知名厂商：腾讯、网易、完美世界、巨人网络等。

4. 嵌入式和物联网领域

嵌入式：就是把具有计算能力的主控板嵌入到机器装置或者电子装置的内部，能够控制这些装置。比如：智能手环、摄像头、扫地机器人、智能音响等。

谈到嵌入式开发，大家最能想到的就是单片机开发(即在8位、16位或者32位单片机产品或者裸机上进行的开发)，嵌入式开发除了单片机开发以外，还包含在soc片上、系统层面、驱动层面以及应用、中间件层面的开发。

常见的岗位有：嵌入式开发工程师、驱动开发工程师、系统开发工程师、Linux开发工程师、固件开发工程师等。

知名的一些厂商，比如：以华为、vivo、oppo、小米为代表的手机厂；以紫光展锐、乐鑫为代表的芯片厂；以大疆、海康威视、大华、CVTE等具有自己终端业务厂商；以及海尔、海信、格力等传统家电行业。

随着5G的普及，物联网(即万物互联，)也成为了一种新兴势力，比如：阿里IoT、腾讯IoT、京东、百度、美团等都有硬件相关的事业部。

5. 数字图像处理

数字图像处理中涉及到大量数学矩阵方面的运算，对CPU算力要求比较高，主要的图像处理算法库和开源库等都是C/C++写的，比如：OpenCV、OpenGL等，大名鼎鼎的Photoshop就是C++写的。

6. 人工智能

一提到人工智能，大家首先想到的就是python，认为学习人工智能就要学习python，这个是误区，python中库比较丰富，使用python可以快速搭建神经网络、填入参数导入数据就可以开始训练模型了。但人工智能背后深度学习算法等核心还是用C++写的。

7. 分布式应用

近年来移动互联网的兴起，各应用数据量业务量不断攀升；后端架构要不断提高性能和并发能力才能应对大信息时代的来临。在分布式领域，好些分布式框架、文件系统、中间组件等都是C++开发的。对分布式计算影响极大的Hadoop生态的几个重量级组件：HDFS、zookeeper、HBase等，也都是基于Google用C++实现的GFS、Chubby、BigTable。包括分布式计算框架MapReduce也是Google先用C++实现了一套，之后才有开源的java版本。

除了上述领域外，在：科学计算、浏览器、流媒体开发、网络软件等都是C++比较适合的场景，作为一名老牌语言的常青树，C++一直霸占编程语言前5名，肯定有其存在的价值。

3.3 在校招领域

3.3.1 岗位需求

校园招聘

毕业时间: 2021年9月1日 - 2022年8月31日 【中国大陆（内地）以毕业证为准，中国港澳台及海外地区以学位证为准】
网申截止日期: 2021年9月27日24:00（北京时间）

腾讯2022校园招聘启动公告&FAQ

软件开发-后台开发方向

岗位描述

你是腾讯产品「背后」的英雄，
服务后台的架构设计、开发、优化以及运营是你的绝学；
你通过网络接入、业务运行逻辑、用户数据存储、业务数据挖掘等方向的优化，
打造出更稳定、安全、高效和可靠的专业后台支撑体系，
守护海量用户的笑容，
深藏功与名。

岗位要求

必须具备的：

扎实的编程能力；

C/C++/Java开发语言；

TCP/UDP网络协议及相关编程、进程间通讯编程；

专业软件知识，包括算法、操作系统、软件工程、设计模式、数据结构、数据库系统、网络安全等。

有一定了解的：

Python、Shell、Perl等脚本语言；

MySQL及SQL语言、编程；

NoSQL、Key-value存储原理。

可以加分的：

分布式系统设计与开发、负载均衡技术，系统容灾设计，高可用系统等知识。

算法能力要强，有一定项目开发经验

至少要掌握一门面向对象的语言，C++或者java都可以

要熟悉网络编程

网易 校园招聘
NETEASE

首页

互联网职位

人才专项

雷火游戏职位

互娱游戏职位

走进网易

C++开发工程师 🔥 紧缺

岗位描述

- 1、俗话说万丈高楼平地起，所以产品服务器端开发就交给你；
- 2、俗话说每个程序员都是自己代码的秘书，所以你要根据产品需求完成服务器编码、测试以及文档编写工作；
- 3、俗话说服务器就是战斗堡垒，所以你要维护线上部署服务，保证服务器质量和性能；
- 4、俗话说扩充基地是第一要义，所以存储新技术研究、开发也是你的职责，来靠你的聪明才智设计高性价比的存储系统吧；
- 5、俗话说云计算少不了技术支持，所以你要为它提供高性价比的数据库和存储解决方案。

岗位要求

我们希望你：

- 1、不管学的是什么专业，都有一颗对互联网无比热爱，愿意为之付出的心
- 2、就是喜欢C、C++这个程序语言，还就爱拿它做项目开发；
- 3、对网络编程有研究，基本的gdb调试能力？内存泄漏，越界读写等常见问题？对你来说都不是问题；
- 4、作为一个优秀的程序员，有良好的服务器端编程能力和编程习惯，相关模块设计、文档编写都是小case；
- 5、十八般武艺样样精通，掌握多线程并发编程技术，掌握各种数据结构和算法
- 6、熟悉windows/linux编程环境，如果有MySQL开发经验那就再好不过啦！！

3.3.2 笔试题

[2021届奇安信C++方向试卷一](#)

[2021届网易C++工程师提前批](#)

[2021届腾讯C++工程师笔试题](#)

近两年来校招笔试主要变化:

1. 基本上都采用在线oj方式, 成本低, 效率高
2. 企业对代码能力要求越来越高, 代码能力差的同学直接就被比特pass了
3. 中小厂的笔试可能还有选择题, 大厂的笔试一般都是3个以上的算法题, 而且普遍有一定难度

因此大家在学习过程中, 一定要多多敲代码, 坚持刷题!!!

3.3.3 面试题

【某学长深信服一面】

1. 自我介绍
2. 你的云备份项目讲一下, 用了 哪些技术? 【项目】
3. 堆栈申请空间有什么区别? 【C++】
4. 不释放空间 会导致 什么? ---内存泄漏 【C++】
5. 内存泄露的检测? 解决办法? 了解哪些内存泄露检测工具? 【C++】
6. 如果说有 很多的数据, 现在应该使用什么数据结构来保证对这些数据操作的效率? 【数据结构/算法】
7. 讲一下红黑树 【数据结构/算法】
8. 红黑树和 AVL 的区别? 红黑树 最多几次旋转可以保持平衡? 【数据结构/算法】
9. 你了解红黑树的旋转吗? 【数据结构/算法】
10. 写代码, 字符串匹配, 不允许使用 C++, 使用 C语言? 【数据结构/算法】
11. 问题, 现在有大量有序数据, 你能不能设计一个高效的算法, 把这个有序数据随机打乱? 要求: 使用 rand 函数, 说了一个 复杂度 $n \log n$, 问还有没有更好的 想法, 进行改进? 【数据结构/算法】

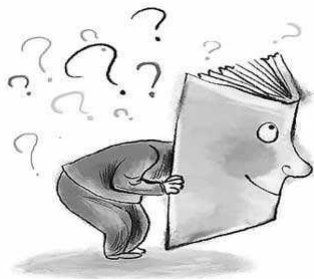
【某学长腾讯面经】

1. 自我介绍
2. 实习项目 【项目】
3. C++ 面向对象特性, C++ 中多态实现机制, 虚继承是如何处理二义性的问题的? 【C++】
4. 智能指针 【C++】
5. TCP三次握手、四次挥手, TCP 如何实现可靠传输? 【网络】
6. 当三次握手第三次报文丢掉时, 此时服务器端、客户端分别处于什么状态? 【网络】
7. 为什么要等待2MSL? 如果有大量的连接处于TIME_WAIT状态, 有何影响? 【网络】
8. UDP如何实现可靠传输? 【网络】
9. mysql数据库存储引擎? B+树比B树好在哪里? 数据库事务及特性 【数据库】
10. 进程间通信方式, 进程和线程的区别 【操作系统】
11. 虚拟内存到物理内存如何映射的? 常见的页面置换算法有哪些? 【操作系统】
12. map, unordered_map区别和应用场景 【C++】
13. vector、list区别? 【C++】
14. 算法题: 反转单链表 【数据结构/算法】
15. 算法题: 二叉树的最大路径和 【数据结构/算法】

从校招中公司岗位的技能要求, 以及学长面经总结了解到, 公司在校招期间更看重学生的基础, 最主要是: 语言(至少掌握一门面向对象语言java/C++)、数据结构、操作系统、网络、数据库、设计模式等, 而本门C++的授课内容, 更注重学生的实践动手能力、工作中的应用以及笔试面试中的技巧, 最后达到能够正常工作以及学习即可。

4. 如何学习C++

学习C语言需要多多练习，学习数据结构需要多多思考，那学习C++呢？



4.1 别人怎么学？

以下引用自2010年8月号《程序员》刊登的拙文《C++强大背后》最后一段：

C++缺点之一，是相对许多语言复杂，而且难学难精。许多人说学习C语言只需一本K&R《C程序设计语言》即可，但C++书籍却是多不胜数。我是从C进入C++，皆是靠阅读自学。在此分享一点学习心得。个人认为，学习C++可分为4个层次：

- **第一个层次，C++基础** (平平常常)

挑选一本入门书籍，如《C++ Primer》、《C++大学教程》或Stroustrup撰写的经典《C++程序设计语言》或他一年半前的新作《C++程序设计原理与实践》，而一般C++课程也止于此，另外《C++ 标准程序库》及《The C++ Standard Library Extensions》可供参考；

- **第二个层次，正确高效的使用C++** (驾轻就熟)

此层次开始必须自修，阅读过《(More)Effective C++》、《(More)Exceptional C++》、《Effective STL》及《C++编程规范》等，才适宜踏入专业C++开发之路；

- **第三个层次，深入解读C++** (出神入化)

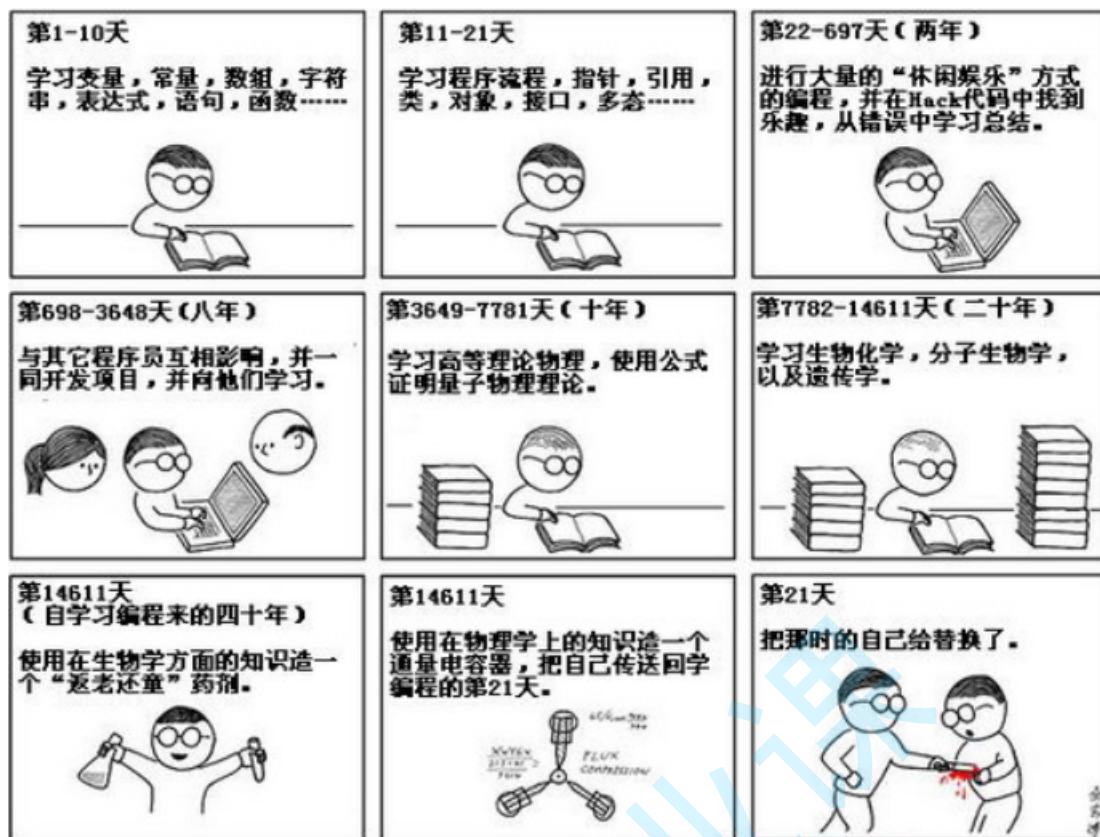
关于全局问题可读《深入探索C++对象模型》、《Imperfect C++》、《C++沉思录》、《STL源码剖析》，要挑战智商，可看关于模版及模版元编程的书籍如《C++ Templates》、《C++设计新思维》、《C++模版元编程》；

- **第四个层次，研究C++** (返璞归真)

阅读《C++语言的设计和演化》、《编程的本质》(含STL设计背后的数学根基)、C++标准文件《ISO/IEC 14882:2003》、C++标准委员会的提案书和报告书、关于C++的学术文献。由于我主要是应用C++，大约只停留于第二、三个层次。然而，C++只是软件开发的一环而已，单凭语言并不能应付业务和工程上的问题。建议读者不要强求几年内“彻底学会C++的知识”，到达第二层左右便从工作实战中汲取经验，有兴趣才慢慢继续学习更高层次的知识。虽然学习C++有难度，但也是相当有趣且有满足感的。

<http://www.zhihu.com/question/23933514>

如果题主认为的「精通」是指上述所指的第二个层次，那么我估计一年全职时间也未足够「精通」。首先，阅读这些书籍本身也需要不少时间，而且对于一般人来说（指除了一些学习记忆能力超强的人），很可能需要阅读几遍才能记住一些细节。另外，**阅读后必须要练习**。除了入门的习题外，还需要做一些工程向的项目，才能了解各种语法、编程范式的使用方法及时机，理想地建议参与一些开源项目。然而，**学习是需要不断思考并把结果沉淀**，方法包括与朋友讨论、写博客、写知乎答案等，这些都需要时间。



这是迄今为止
我所知道的最容易的
《Teach Yourself C++ in 21 Days》

建议不要把「精通C++」作为一个一年目标，应该要把学习语言作为一个持续的过程，同时要把语言运用在具体的应用场合中。

["21天学会C++"](#)

[“编程能力与编程年龄”](#)

4.2 自己怎么学

虽然C++语法特性繁琐复杂，经典书籍琳琅满目，但是只要你紧跟我们的节奏，你也可以“前途光明”。

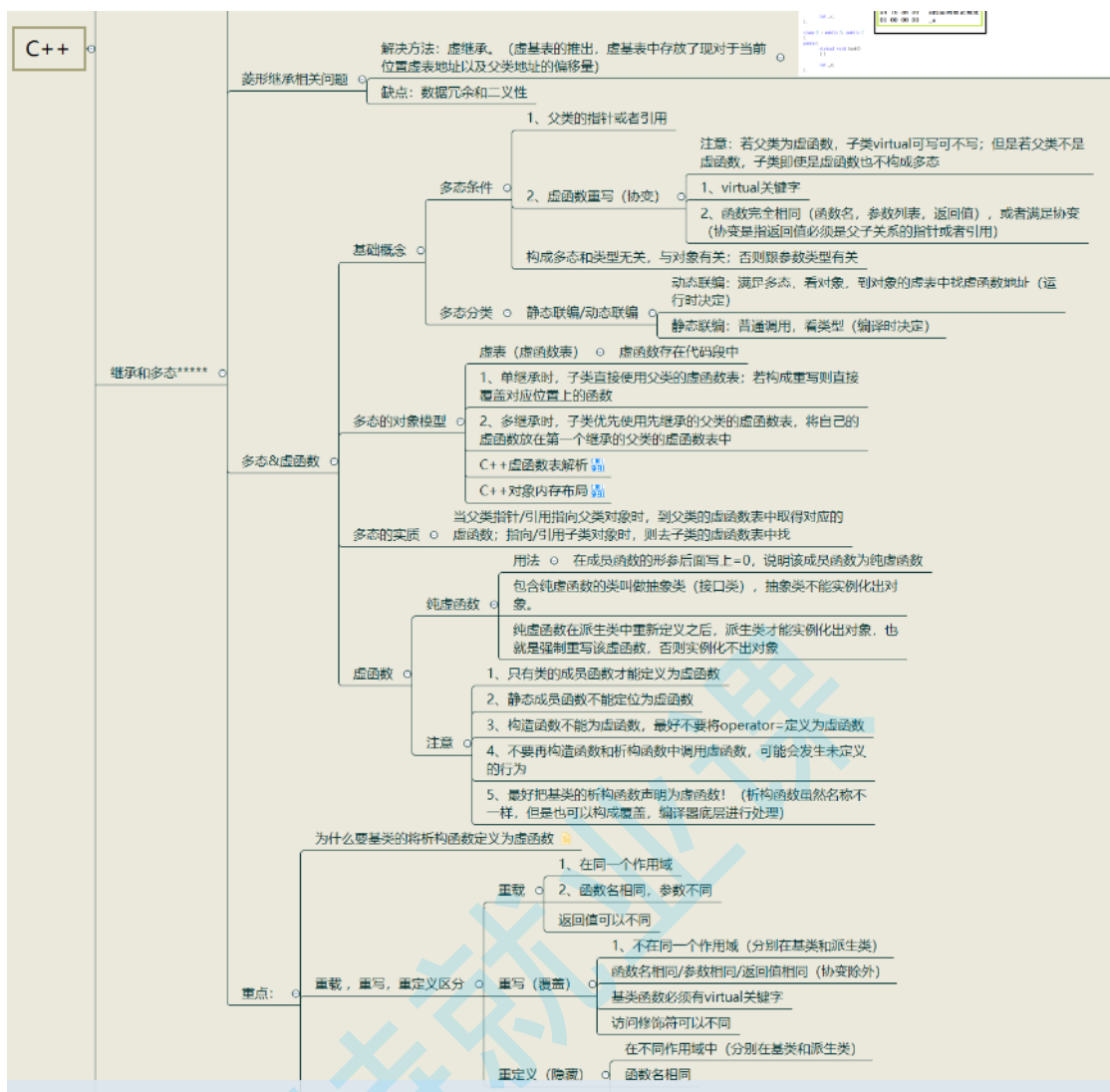
• 多总结

1. 写博客

经常写博客不仅可以记录自己学习的历程，知识点的认识，更是复习的一把神兵利器。

[往届大佬学长博客](#) (先登录教务系统，再点击这个链接就能看)

2. 中后期画思维导图总结



画思维导图软件比如: xmind、imindmap等。

• 常看书

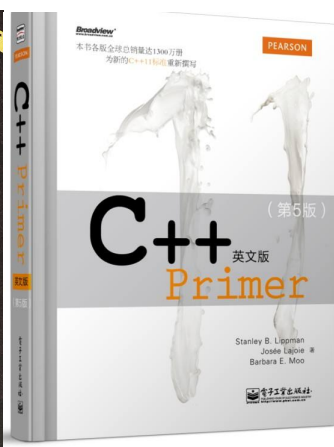
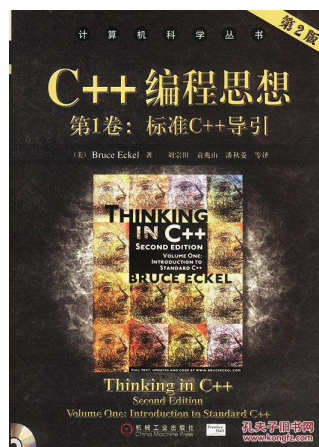
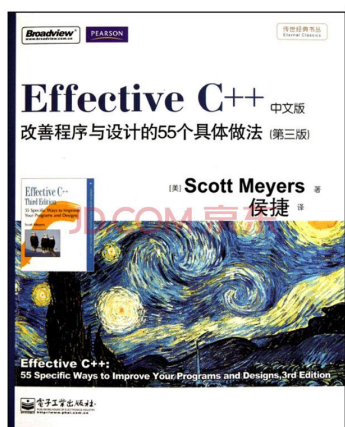
合理的看书安排, 切记贪, 缓缓图之, 否则容易走火入魔。

1. 初窥门径



入门阶段, 找一本简单、薄一点的书快速入门, 《C++程序设计》就挺适合的, 可以简单快速了解C++的语法特性。

2. 潜心修行

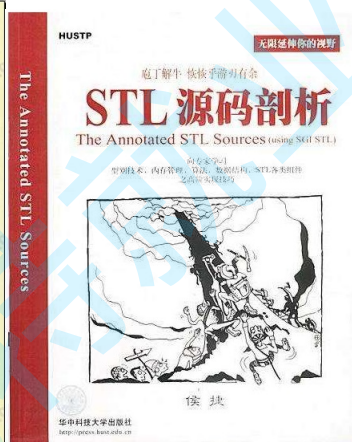
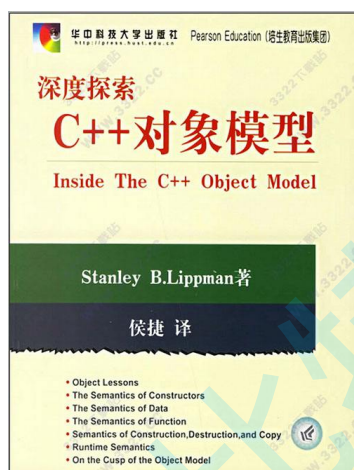


当C++学习过半之后，大概继承和多态学完，建议看下《Effective C++》，该书以建议的方式告诉你如何写出更好的C++程序，看和不看差别非常大。

C++课程系统学完后，有时间可以看下《C++编程思想》对讲过的知识点融会贯通，该本书有一定深度，从原理方面对C++语法进行介绍。

《C++ Prime》可以当成工具书对待，有不清楚的语法特性可以快速去查，建议工作之后可以利用闲暇时间慢慢品味下。

3. 深入浅出



这两本书都有一定难度，如果基础知识掌握不扎实，不建议看，强烈建议好好把基础知识打扎实。

《STL源码剖析》将六大组件逐一剖析，看完之后对STL的设计框架以及内部实现原理有更深入的认识，面试时可以游刃有余和面试官交锋。

《深度探索C++对象模型》会让你对C++语法设计原理有更深层次的认识，有些疑惑的知识点看完之后里面就清晰了。

• 勤刷题

1. 牛客网

[牛客网专项训练](#)

2. leetcode

[leetcode中文刷题网站](#)

5. 关于本门课程

本门课程不仅仅是C++语法的学习，我们的目标是让大家如何高效的正常工作，因此本门课程涉及到以下内容：

- C++的基本语法
- STL库
- 高阶的数据结构

C++不仅仅是一门语言，更是一种生活哲学，在以后的工作之余还需要大家继续挖掘.....



比特就业课