C++书单

第一篇

一些同学问我，如何学好C++，我没有别的办法给你们，唯一的办法就是读书，读大量的书，就可以解决。要把C++作为日常语言，而不是一种程序语言，这样就好办了。

有人又要问我，那么我应该读什么书才好？没有时间怎么办？我只能对你们说，没时间的话，就别学C++了，做你们喜欢做的事。生活中没有C++，也同样美好。如果你准备学，一定要学好，那么我开个书单，应该问题不是甚大。

首先肯定要读一读Bjarne Stroustrup的The Design and Evolution of C++，了解

一下这个语言的历史。接下来就可以看别的书了，但要不停地回头看这本书，看到你不断地学到的新技术是怎么样一点点地被接纳到这个语言中去的。

第一本书因人而异，基础好一些的，可以看Stanley B. Lippman的C++ Primer，这本书非常地巨大，你打星号的部分可以不要看。基础不太好的，可以看Stanley B.Lippman的Essential C++，这本书份量要轻得多，不过四个C++的范型都讲了，而且讲得非常清楚。

第二本应该停止技术层面的东西，静下心来看看Pike和Kernighan的The Practice of Programming，好好地整理一下，在程序设计中应该有哪些注意的事项。这本非常薄的booklet，可以说是程序员必读的指南。

第三本书，就应该是Bruce Eckel写的、候捷译的Thinking in C++，这本书每过半年我就要重读一遍。可以说每一章都是写得发人深省的，这本书让我感觉到了技术运用的非常高的境界，但是语言非常平实，只要认真地读，即使基础不行，也一定可以懂。我在教课的时候，就是用这本书（面对的学生是零基础）。

要更上一层的话，就要慢一步，先要把握C++设计习惯的良好。这是Scott Meyers的Effective C++和More Effective C++带给我们的无尽收益。我More Effective C++买不起，只好花了10块钱复印装订了一本"线装本"，看起来像葵花宝典。这两本书是真正的经典，作者对C++的纯熟，使得语言的风格读起来简直是如饴甘甜，就像他站在对面在讲课。我手中有这两本书的原版CD，如果有兴趣，可以发E-mail到sjtu@263.net或在饮水思源投条儿给gaobo索要，只要您提供光盘我就给免费烧。如果你已经深刻地理解了Effective C++和More EffectiveC++，那你可以发现，你在众人中已经是鸡群之鹤。可以指导项目运作了，可以编写一切你想做的程序了，可以指出别人看起来不错的代码的大小问题了。如果你能一眼看出有人的代码是对应于"条款27"或"条款M6"，那你可真是让本人刮目了。

我已经讲了，如果要写程序，EC++和MEC++的境界已经足以使你自如应付，可是如果你还不满足，想关注一些理论层面的问题，或是想看看实现的代码，你就不应该错过这几本好极了的书。我是说Herb Sutter的Exceptional C++和More Exceptional C++，这两本书的难度是非常大的，我对每一条的阅读笔记都是十多页。特别是泛型程序设计的部分，这两本书旁征博引，极尽深入探讨之能事，每每看懂一条，都抹汗一次，大感酣畅淋漓；还有侯捷的　STL源码剖析，以实际的例子一点点地讲解一个STL是怎么样实现的，我是刚开始读，不发表评论；而Stanley B. Lippman，Cfront的实现者之一，执笔写出Inside the C++ Object Model，我只有一个字，就是基本帅呆了。我从中了解了无数的编译器解释源代码的细节，以及记忆体分配的细节，呵呵，这些都知道了，我还怕什么呢？最近得到了另一Cfront实现者、C++标准委员会Koenig的C++沉思录，看起来非常不错，这里也推荐给大家，但我也没看完，亦无发言权。

最后最后，你们，未来的C++理论家们，可要记住，Bjarne Stroustrup的The C++ Programming Language无论如何也应该读个四五遍！这是一切C++的书本的源泉。

如果还觉得不够，就向C++标准委员会订购一本C++标准。

一切中国大陆作者的书，一概不要看（包括我的）。一切VC++或讲特定的编译器的书，一概不要看。如果需要补C语言的课，买一本非常小的K&R的The C Programming Language足矣，其它的书一概不要看。不要先学C，再学C++，而要直接学C++。你不是先学古文，再学白话的，对不对？所以相信我，直接来更容易。

以上文字，皆为原创，本人愿意为每个字负责。

第二篇

【阶段一】：

《C++程序设计语言（特别版）》

The C++ Programming Language (Bjarne Stroustrup)

《C++语言的设计和演化》

《The Design and Evolution of C++》

《C++ Primer plus (3RD)中文版》

《C++ Primer plus(4RD)中文版》\*\*\*

《Essential C++中文版》

《Accelerated C++中文版》

《Thinking In C++》（不要看中文版）\*

【阶段二】：

《Effective C++中文版》

《More Effective C++中文版》

《Exceptional C++中文版》

《More Exceptional C++中文版》

《C++沉思录》

【阶段三】：

《深度探索C++对象模型》

Inside The C++ Object Model (Stanley B. Lippman)

《设计模式：可复用面向对象软件的基础》

Design Patterns

【STL】:

《C++ Templates全览》

《C++设计新思维：泛型编程与设计模式之应用》\*\*\*

《C++标准程序库：自修教程与参考手册》

STL Tutorial and Reference Guide （David R. Musser)

《Effective STL》

【网络】

《C++网络编程，卷1：运用ACE和模式消除复杂性》

《C++网络编程，卷2：基于 ACE 和框架的系统化复用》

第三篇

C++学习推荐书目

语言入门:

有丰富C经验,缺乏C++经验的: 《C++ Primer (3RD)》 ；有其他语言的丰富经验的: 《The C++ Programming Language》简称TCPL(对不住纯C的fans了) ；如果选择薄一点的书：《Essential C++》《Accelerated C++》都是不错的选择。 其实，我个人一直认为C++Primer带着太过强烈的C的痕迹，对于学习C++未必是好事。 Accelerated C++我没有读过，从各方面的评价来看，完全值得推荐。

这几本书都有相应的中文版，而且翻译的质量都不错。上面的书未必都需要读一遍，但无论如何，TCPL是应该阅读的。

进阶A：

这个路线侧重于语言本身

《Effective C++》 简称EC，C++程序员必读！如果我有权招聘程序员的话，没有读过这本C++“圣经”的人，大概...很多时候，我们说C++圣经不是指TCPL，而是这一本。

《C++ Coding Standards: 101 Rules, Guidelines, and Best Practices》简称《C++编程规范》，个人认为，这是C++程序员必备的案头书。几乎Effective系列和Exceptional系列都在这里得到了总结。最新的模版、异常的业界经验都在这里的到了体现。可能的唯一缺陷就是对一个新手而言，对于为什么这么做，解释的不够。我的看法是：如果你不理解其中的条款，记忆，并且照做；如果你理解其中的条款，我猜你一定会同意书中的观点。我估计，这本书中的内容至少在2009年以前都不会过时，人们将广为传诵它制定的101条戒律。还不知道他的简称，也许101会成为一个候选者？

提到《Effective C++》，那么另外三本书一一浮出水面：

《More Effective C++》简称MEC，《Exceptional C++》和《More Exceptional C++》，新书《Exceptional C++ Style》也是值得一看的好书。 上述几本书，一本也不应该放过，个人建议上述书籍按顺序阅读。并且，在将来反复阅读上述几本书。

进阶B：

这个路线侧重于标准程序库。

《The C++ Standard Library》听说过STL吗？这本书会教你最基本的，也是最重要的STL的使用。

STL编程指南(Programmer's Guide)电子书http://www.shubulo.com/thread-35929-1-1.html

中文版名称：《C++标准程序库 自修教程与参考手册》，这本书，应该是必读的。

《Effective STL》会使用STL还不够，还必须知道什么时候选择什么STL组件，这是STL使用的必读书。

《Generic Programming and the STL: Using and Extending the C++ Standard Template Library》，中文名称《泛型编程与STL》，这本书理论性较强，但是真的很严谨，而且并不是非常难懂。理解其中对于Concept的解释，将是非常重要的。

《C++ STL》这不是讲述使用程序库的，而是讲述程序库实现原理的。肠胃不好的，需要慢慢吸收。是不是翻译的问题呢？ 这个路线的书，仍然是建议按顺序阅读。

AB两个路线应该是可以并行的。

其他进阶参考书

提出这些参考书，只是避免争议，我只认真看完了其中的一本。

《Thinking in C++》这本书及其中文版传言好坏都有，没有认真看过，不做评价，如果确有兴趣，不妨尝试一下该书。

《C++必知必会》，如果早一年，这本书将是重量级的，然而它被101和《Exceptional C++ Style》盖过一头。

《C++ Gotchas: Avoiding Common Problems in Coding and Design》，这是一本我未曾读过的书，而且广泛的获得好评。中文版《C++程序设计陷阱》

《STL 源码剖析》这本书我刚到手，就被人“借”走，以至于我到现在也没有看过。看过这本书的朋友，可以给一个合适的评价。

《Imperfect C++》，这确实是一本好书，个人认为不大适合缺乏工程经验的人。不过，如果你看完了前面步骤的书，那么看这一本书将是一件非常轻松的事情。书中光盘所附的stlsoft库还真的是一个精品。到目前为止，这本书我刚看了一半。

高级：

《The Design and Evolution of C++》简称D&E，内容并不艰深，Bjarne的书，仅此，就值得一读。

《Inside The C++ Object Model》中文版：《深度探索C++对象模型》，这个C程序员一定会觉得C++原来并不神秘。

《C++ Template》在阅读STL、Boost或者Loki的源代码之前，请仔细阅读本书，它可以减轻一些阅读的烦恼。这本书是谈论C++模版的权威。

《Modern C++ Design Generic Programming and Design Patterns Applied》简称MCD，中文书名《C++设计新思维：泛型编程与设计模式之应用》。在阅读MCD之前，简易先阅读C++ Template。

你对C++觉得不满吗？《对象揭秘：Java、Eiffel和C++》可以部分的帮助你完成抱怨的目标。也许它还可以让你不迷信C++。

另外，一份C++标准文档也应该是加以咀嚼的。

还有一些书籍，并不能简单的归于C++，也难以在纯粹的书本学习中加以掌握。《设计模式》一书就是这样。因此，这里推荐的仅仅限于C++语言本身。

所有上述的书籍，不是谈论C++语言本身，就是谈论STL的。偶尔，某些书中条目也会涉及问题相关的现实工程。这些书并不是C++软件开发的全部，但是他们很重要。阅读这些书，未必需要化费太多的精力，有时候会困难，有时候会很快速。回顾我自己阅读上述书籍的经历，这差不多是3年的时间。

有几本书，需要说明一下的。

《Multi-Paradigm Design For C++》这本书有些陈旧，也许在出版的年代还是很优秀的，现在，这样的书已经完全可以被推荐书目覆盖了。最重要的是，书中的某些观点今天已经不合时宜了。

《C++编程惯用法》这是一本如雷贯耳的书，可惜，这本书是1992年出版的，书中的很多内容，已经被后来的作品覆盖，而且讲述得更全面严谨。另外，该书中对于C++的认识，在今天看来，已经过时甚至成为错误的观点。因此，只适合已经通晓C++的人参考。

《深入浅出MFC》这是一本讲述如何使用MFC库的书。今天，MFC日渐式微，而且，MFC所使用的C++技术也是一直为人所诟病的。书确实是不错的，但是，学习C++并不一定要学习MFC。因此，对于学习C++而言，这本书不是必须。

《The Art Of C++》老实说，我对这本书的评价一般般。而且，“Art”名不副实，有翻译成《C++实战》，到是更确切一些。

《大规模C++编程》这是一本重要的书，特别是对于缺乏大规模C++软件开发的人来说，尤其如此。但是，这本书的中文版翻译太差劲了。这是一本关于工程的书，因此，并不是C++学习必备的。另外，书中的观点略嫌保守，对于最新的C++技术运用不多。