

《Accelerated C++ 中文版》

图书基本信息

书名：《Accelerated C++ 中文版》

13位ISBN编号：9787111224044

10位ISBN编号：7111224043

出版时间：2008-1-1

出版社：机械工业出版社

作者：（美）克尼格（Koenig, A.），（美）莫（Moo, B.E.）

页数：293

译者：靳志伟

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

内容概要

《Accelerated C++ 中文版通过示例进行编程实践》系统介绍C++程序设计，是美国斯坦福大学的经典教材。从使用C++标准库中的高级抽象开始，使读者很快掌握编程方法。每一章都有很经典独特的例子以及非常到位的讲解，覆盖了C++更多领域的内容，从标准库容器、泛型算法的使用，到类的设计、泛型算法的设计，本书都进行了详细的讲解。

本书作者有丰富的C++开发、研究和教学经验，内容由浅入深，讲解精炼巧妙。无论是刚入门的新手还是有经验的C++开发人员都能从本书中受益。

作者简介

Andrew Koenig 和 Barbara E. Moo 堪称 C++ 研究领域的“第一神仙眷侣”，他们不光有着多年的 C++ 开发、研究和教学经验，而且亲身参与了 C++ 的演化和变革，是对 C++ 的变化和发展起到重要影响的人。

Andrew Koenig，AT&T 大规模程序研发部（前贝尔实验室）成员，同时也是 C++ 标准委员会项目编辑。他有 30 多年编程经验，15 年 C++ 开发经验，已发表 150 多篇与 C++ 有关的论文，应邀在世界各地多次演讲。曾编著 C Traps and Pitfalls 一书，并与妻子合著 Ruminations on C++。

Barbara E. Moo，独立咨询顾问，在软件领域从业 20 多年。在 AT&T 工作的近 15 年中，参与了第一个使用 C++ 编写商业产品的开发项目，负责管理公司第一个 C++ 编译器项目，并成功指导开发了 AT&T 中屡获殊荣的 WorldNet Internet Service Business。曾参与编写 Ruminations on C++ 一书，并在世界各地进行过多次演讲。

书籍目录

第0章 开始

- 0.1 注释
- 0.2 #include指令
- 0.3 主函数main
- 0.4 花括号
- 0.5 使用标准库进行输出
- 0.6 返回语句
- 0.7 进一步的深入
- 0.8 小结

练习

第1章 字符串的使用

- 1.1 输入
- 1.2 将姓名装框
- 1.3 小结

练习

第2章 循环与计数

- 2.1 问题
- 2.2 程序的整体结构
- 2.3 输出数目未知的行
- 2.4 输出一行
- 2.5 完整的框架程序
- 2.6 计数
- 2.7 小结

练习

第3章 使用批量数据

- 3.1 计算学生成绩
- 3.2 用中值代替平均值
- 3.3 小结

练习

第4章 组织程序和数据

- 4.1 组织计算
- 4.2 组织数据
- 4.3 将各部分代码连接到一起
- 4.4 将计算成绩的程序分块
- 4.5 修正后的计算成绩程序
- 4.6 小结

练习

第5章 使用顺序容器和分析字符串

- 5.1 将学生进行分类
- 5.2 迭代器
- 5.3 用迭代器代替索引
- 5.4 重新思考数据结构以实现更好的性能
- 5.5 list类型
- 5.6 分割字符串
- 5.7 测试split函数
- 5.8 连接字符串
- 5.9 小结

练习

第6章 使用库算法

6.1 分析字符串

6.2 比较计算成绩的方案

6.3 对学生进行分类并回顾我们的问题

6.4 算法、容器以及迭代器

6.5 小结

练习

第7章 使用关联容器

7.1 支持高效查找的容器

7.2 计算单词数量

7.3 生成交叉引用表

7.4 生成语句

7.5 关于性能的一些说明

7.6 小结

练习

第8章 编写泛型函数

8.1 什么是泛型函数

8.2 数据结构独立性

8.3 输入和输出迭代器

8.4 使用迭代器提高适应性

8.5 小结

练习

第9章 定义新类型

9.1 Student_info回顾

9.2 自定义类型

9.3 保护

9.4 Student_info类

9.5 构造函数

9.6 使用Student_info类

9.7 小结

练习

第10章 管理内存与低级数据结构

10.1 指针与数组

10.2 字符串字面量回顾

10.3 初始化字符串指针数组

10.4 main函数的参数

10.5 文件读写

10.6 内存管理的三种方法

10.7 小结

练习

第11章 定义抽象数据类型

11.1 Vec类

11.2 实现Vec类

11.3 复制控制

11.4 动态的Vec类型对象

11.5 灵活的内存管理

11.6 小结

练习

第12章 使类对象获得数值功能

12.1 一个简单的string类

12.2 自动转换

12.3 Str操作

12.4 有些转换是危险的

12.5 类型转换操作函数

12.6 类型转换与内存管理

12.7 小结

练习

第13章 继承与动态绑定的使用

13.1 继承

13.2 多态与虚拟函数

13.3 使用继承解决问题

13.4 一个简单的句柄类

13.5 使用句柄类

13.6 微妙之处

13.7 小结

练习

第14章 近乎自动地管理内存

14.1 用于复制对象的句柄

14.2 引用计数句柄

14.3 可以让您决定何时共享数据的句柄

14.4 可控句柄的一个改进

14.5 小结

练习

第15章 再探字符图形

15.1 设计

15.2 实现

15.3 小结

练习

第16章 学习C++的后续方法

16.1 利用已经掌握的知识

16.2 学习更多的知识

练习

附录A C++语法细节

A.1 声明

A.1.1 指定说明

A.1.2 声明符

A.2 类型

A.2.1 整数类型

A.2.2 浮点类型

A.2.3 常量表达式

A.2.4 类型转换

A.2.5 枚举类型

A.2.6 重载

A.3 表达式

A.4 语句

附录B 标准库一览

B.1 输入-输出

B.2 容器和迭代器

B.2.1 共有的容器操作

B.2.2 顺序容器的操作

B.2.3 其他顺序操作

B.2.4 关联容器的操作

B.2.5 迭代器 (iterator)

B.2.6 向量 (vector)

B.2.7 链表 (list)

B.2.8 字符串 (string)

B.2.9 对 (pair)

B.2.10 图 (map)

B.3 算法

精彩短评

- 1、 这本书是我的第一本C/C++ 正式学习书籍，之所以说是正式，是因为原来看过谭浩强的C++，以及一本C++简读本的书，发现当时浪费了大量时间，如果刚开始就这本加速C++学起来，效果可能会更好，这本书的确写得非常好，作者从初学者角度将例子贯穿与知识点，教学方法非常独特，相比与国内很多只是写一堆理论，然后给几个例子和习题来说，作者在这本书的写作上用了很多的功夫来思考，将这本书从头到尾看完，你将会第一步感受到一个C++的概貌，当然，这本书又比较薄，自然会有很多细节知识点和惯用法没有涉及到，但是，学习的本质不就是掌握核心理论，然后在实践中不停的补充新的知识么，所以推荐这本书看过去学习经典C++牛人的书The C++ Programming Language，目前正在钻研，感觉写的很好，如果在学习的过程中能够接触实际的项目锻炼，提高会更加的快，希望大家都能学好用好C++，为自己的具体实践工作打好基础
- 2、 这是一本杰出的书，对于标准库的使用和内存管理的介绍精准而有效率。
- 3、 快速的掌握 C++ 的主要知识 最有用的部分
以 80% 场景为主 非常适合入门 不被细节所拖累
- 4、 太赞了！！正在看这本书
- 5、 亲，你看的哪个翻译版本？？覃剑锋 or 靳志伟
- 6、 换一种角度学习C++，眼界就变得开阔起来
- 7、 我发现国外的大牛，程序写的牛，文章写得好，这再一次证明：好的程序员一定是个好的作家，即便不是，那也是没人挖掘其潜力！
此书确实极力避免了指针，讲述方式也比较新颖，作为一个学过C++，至少写过几千行有用程序的人来讲，还是有点吃力。
建议初学者，还是找本薄薄的c++先看，再拿那本primer作为参考手册，然后再看此类书籍了。。
老实说，现在读书的人少了，尤其是学生：本应该打基础，多开阔眼界之时，结果都被用于花前月下，dota睡觉中了，不是不能娱乐，而是现在的很多学生已经除了娱乐外基本不学习了，学习？那只是为了考试。
不吐槽了，看书看书.....
- 8、 中文翻译读不下去。这书你好好学过C就不会觉得太累。说知识浓度高，无非就是把许多坑一句话带过了，说实话这样写书很不友好，也难怪primer那种繁琐的书评价那么高。不是很推荐看。
- 9、 <http://www.parksc computing.com/accelerated-cpp-solutions/>

上面鏈接的答案還沒有寫完，若還需要搜某題的答案，可使用 " Accelerated C++ exercise 3-2 " 這個格式搜。

以上是在 <http://www.cnblogs.com/logitechlike/archive/2012/01/04/2312589.html> 上看來的。

5.2補充

評論中@Rockingpepper君找到了本書全部答案：

<https://github.com/bitsai/book-exercises/tree/master/Accelerated%20C%2B%2B>

- 10、 嗯，正在读，感觉还不错
- 11、 还行吧，一上来就是面向对象编程，纯c的过度有些不适应。适合新手入门
- 12、 Koneing两口子真不愧为教育专家。从20多年前的《C陷阱与缺陷》，到《C++沉思录》，再到这本《Accelerated C++》无不是语言学习的必修之课。特别这本《Accelerated C++》可以说是最佳的C++入门书籍。
Koneing对C++学习有三点建议：尽量避免使用指针；提倡使用程序库；用类表达概念。在这本书中充分体现了这几条原则。整本书的重点放在标准库的规范使用上，后面提到的类的设计，而对于指针基本没有涉及。
指针是C语言最重要的内容，这个特性也继承给了C++。但经过20多年的应用，现在的C++早不

再是C的扩充，它还加入OO、GP等特性了。C和C++早就应该作为两门不同的语言加以学习，Bjarne Stroustrup同样也是这样认为。我们要清醒的认识到：C++作为现今最复杂的一门程序设计语言，在入门的时候不可能面面俱到，选择是困难的。还好，在我们没有很好的主意的时候Koneing夫妇帮我们作出了选择。我认为如果不是将C++应用于一些特殊的领域，这样的入门不论对学习者还是教师都是让人比较愉快的。

13、大一，程序设计实践

14、是的，在c traps and pitfalls中，作者就说过在c++中少用指针。

15、这是我通读完的第一本C++教程，之前只读过primer的前几章。

不同于其他那些砖一样的入门书籍，Accelerated C++只有短短不到300页，而又是以实例开始逐步深入，对于初学者而言是莫大的福音。然而，要想读透本书，完成所有课后习题，绝非急功近利者短期能够完成的。所以，用深入浅出来形容Accelerated C++再恰当不过了。

作为初学者，个人以为此书也并非完美。不知道是我没读透的缘故还是翻译的问题，书中的一些描述（包括少数习题）令人十分费解，相关部分重读多遍依然无法理解作者的含义。个人的看法，如果英语了得，尽量还是读原版。

没有习题答案也是件尴尬的事情。做不出来的自不消说，有些题目费劲力气、勉强完成又不知道是否真正达到了题目的要求和作者出题的目的。网上对此多有抱怨。。。作为一个初学者，这个恐怕只有多读两遍再看了。

16、一开始就从vector入手讲解库函数的使用，没有接触过c/c++的初学者，不了解c/c++的数据类型和基本概念看起来可能有些费事。

整本书很紧凑，围绕着几个案例进行，讲解很清晰。

17、终于读完了第一遍，当然懵懵懂懂地，不过真有种醍醐灌顶的感觉。大牛的书写的就是不一样，各种抽象一点点展示给你看，应该怎么做，为什么要这么做讲解得很明白。看完这本书，再看那些先讲C再以C+面向对象方式讲解C++的书，简直就是垃圾。C++完全就是一门全新的语言，OOP以及种种抽象概念才是其精髓。最后要说的是，这本书不太适合初入门来看，讲解的是很详细，但C++太庞杂了，虽然自己有C的基础，也熟悉使用过fortran/matlab，也能看懂Python，看这本书还是比较吃力，各种概念一个接一个来。不过，考虑到C++书要么大部头，要么先讲C，从头培养一个很好的概念/抽象理念还是很重要的，所以如果实在找不到好书，这书做为入门也未尝不可，只是在读之前，一定要告诉自己：坚持读完！之后，你绝对会受益匪浅。

18、这本书真是入门级啊，看得我十分着急。结果后半段讲深入了又不好好看了。。

19、<https://github.com/bitsai/book-exercises/tree/master/Accelerated%20C%2B%2B> 在底下评论里看到的。是全部的答案

20、简洁，但类似于K&R C，初学者到后面就不是很明白一些巧妙的表达式是怎么写出来的了，缺少足够铺陈，还要找别的C++书读。

21、我就是拿来破处的，看到第6章实在看不下去了，看得晕头八脑，习题又做不明白。。。

22、我大一，刚接触c++，这本书对我会不会太难呢？？？

23、这本书作为c++的入门图书太好不过了，不愧是Bjarne Stroustrup推荐的c++入门读物。本书开始就是用标准库的东西来教你如何编写自己的c++程序。尤其是后面步入佳境，vector，string的简单实现，以及后来的通用句柄Ptr来近乎自动管理内存，到后来的字符图形问题的整体设计思路，都使我受益匪浅。不得不说是好书一本！~

24、中文版的翻译一塌糊涂，英文版的短短几个单词就能表达清楚的话，中文版啰嗦半天也词不达意。而且英文版的写的通俗易懂，没有多少生僻词汇。作为在校学生，真替那些在图书馆西文书库里无人问津的英文图书感到可惜。其实英文版的读起来并不比中文版的慢，收获却要比看一本中文书大的多，至少不会得到错误的知识。

25、入门级

26、C++最好的入门书籍，没有之一。把这本书看完，就可以对c++心中有数了

27、 《Accelerated C++》这本书，推荐看英文原版，很有收获，非常好的一本书。中文版偷工减料，有的甚至少翻译一段。有点太不负责任了。有一些问题，翻译的完全搞不明白。这不是害人吗？天地良心！！

28、 国外的教材为何这么好，大牛之笔

29、 算是第一个评论吧，希望给其他人参考。

本来打算评个5星，但是：

1、 翻译一般，基本上抄旧版（覃剑锋翻译的那本，难道是他改名换姓了？），只是换掉了部分词！。某些翻译简直搞笑。

2、 定价过高

3、 排版拙劣，特别是代码排版特差！

ps：针对科学出版社的新版，建议淘宝买旧版或者盗版吧（其他地方好像都缺货，不过英文版倒是有的），谁让正版令我失望呢，国内的出版商实在是不负责任。。。。

30、 请问楼主，像我这种没有一点基础的白纸，自学C++的话，用这本书好吗？

如果不适合，那哪部适合我自学呢？

感谢解答。

31、 由于C++语言的复杂性，在林林总总的C++书籍中，入门的书真的不多。恰好，本书就是属于极易入门的一本书。非常适合初学者阅读。本书循序渐进，个人强烈推荐！

“抱歉，你的评论太短了”。那就来长点==。。。

“抱歉，你的评论太短了”。那就来长点==。。。

32、 学期初的时候买了这本书，又在图书馆借到了相应的中文版，然后就是拿着英文版的一句一句的啃，遇到不懂的地方在看中文的翻译，就这样，半个学期过去了，已经看完了本书的绝大部分。

现在是学期末了，想起来这本书还差倒数第二章，也就是chapter15没有看，因此多少都觉得有点不完美，没有坚持到最后。

不过最后一章chapter16里面有个故事却记得很清楚：

There is an old story about a visitor who has become lost in New York, with tickets in hand to a piano recital. Stopping a passerby, the visitor asks, "Excuse me. Can you tell me how to get to Carnegie Hall?" The answer: "Practice!"

So,现在该做什么呢？

也许就应该像最后一章所建议的那样：重读这本书，做所有第一遍没有做过的练习题！

最后，本书确实是一本入门的好书！

33、 该讲到的都讲到了~

34、 应该是可以的，前提是阁下的C语言比较好了。这本书是作为C语言程序员过渡到C++的快速教程而写的。

35、 太浅显了，不过对于入门者来说，应该是很不错的。条理性不如C++ Primer

36、 我的第一本c++书

37、 我是读过C++ primer将近一年之后才看此书的，比自己预想读完的时间久了很多。这本书每章的信息量都很大，要是自学的话不介意把这本书作文第一本书。但是这本书是值得读的，他没有想Primer那样讲很多的语法，而是讲它最有用的东西，或者说他把C++一个大概的轮廓给你画了出来。我读完之后还是收获很大，之前看的更多的是语法，但是没有实践的运行，而此书则可以在一个实践中把语法中的规定清晰的表达出来，能让你真正的收获到知识。

38、 感觉不如primer看起来条理清楚，可能和机械工业出版社那糟糕的排版有关，都是是大段的

密密麻麻的文字块，看着就累。

39、从C++精要处开始!真好。

翻译很好。

40、@Rockingpepper 太感谢啦!我竟然没注意到,窘:)

41、比起C++ primer那种词典而言,厚度薄了不少,并非像大多数教材一样从C入手,而是采用了C++的思维方式入手,确实不愧是入门教材经典,可惜在国内的知名度还不够

42、这本书,可能是最好的起点之一,当然,绝对不是学习的终点。我在啃了4遍C++ Primer,钻研了几遍effective, more effective,作出了一些小的项目之后,回头来看这本书,觉得本书的内容绝对够资格称得上初学者的圣经。本书的译者,靳志伟先生,在翻译过程中,确实是下了一番功夫,在中国能多有一些这样的翻译,将是一大幸事。但是,我对机械工业出版社,将毫无保留我的憎恨之情,是的,憎恨,你们的排版,纸张,印刷。。。给这么一本经典书抹了黑。能不能反思一下,你们已经把thinking in C这么一本经典给翻译毁了,你们还要错多久?

43、很不错的一本C++入门书籍,建议可以先看国内的C++后,有一点了解在看这本

44、这本书太他妈经典了!!老外写的书真是碉堡了!!

45、覃剑锋

46、读完c++ primer,再读完ac++,惊叹于koenig夫妇竟用如此紧凑的篇幅展示c++诱人的魅力,的确是c++很好的入门书籍.....正像作者所说,“once you have written enough programs that use the material that we have covered so far, you will no longer need the detailed tutorial style that we have adopted in this book. Instead, you will have gained enough C++ programming experience that you will be able to take advantage of books that contain more detailed information, and less explanation, than this one”.,等把这本书题目写完后,果断看the c++ programming language & generic programming and the stl.....

47、我的第一本教程,选它的原因是因为它很不厚,很适合带着到处走,后来认真看了之后发现真的是一本很不错的书,特别是里面的例子很实用(不像其他教科书那样都是要你求两个数的最大公约数之类的问题),并且里面的因为也很易懂,呵呵,推荐像我这样的新手阅读

48、mark一下,没有使用指针的C++?

49、有C基础的人的最佳入门书之一

50、定位很模糊的C++书籍。不如Thinkin in C++和STL源码剖析那样专业,但是内容也不够简单。还好覆盖了大部分C++/STL特性,可以一读。

51、终于读完了第一遍,当然懵懵懂懂地,不过真有种醍醐灌顶的感觉。大牛的书写的就是不一样,各种抽象一点点展示给你看,应该怎么做,为什么要这么做讲解得很明白。看完这本书,再看那些先讲C再以C++面向对象方式讲解C++的书,简直就是垃圾。C++完全就是一门全新的语言,OOP以及种种抽象概念才是其精髓。

最后要说的是,这本书不太适合初入门来看,讲解的是很详细,但C++太庞杂了,虽然自己有C的基础,也熟悉使用过fortran/matlab,也能看懂Python,看这本书还是比较吃力,各种概念一个接一个来。不过,考虑到C++书要么大部头,要么先讲C,从头培养一个很好的概念/抽象理念还是很重要的,所以如果实在找不到好书,这书做为入门也未尝不可,只是在读之前,一定要告诉自己:坚持读完!之后,你绝对会受益匪浅。

52、点评中肯,字字珠玑,入木三分

53、我不喜欢玩游戏,我想学知识,用知识,计算机方面的技术书我看得少,向你学习~我要关注你~第一剑豪~

54、本书不是从"C++中的C"开始讲解,而是始于地道的C++特性.从一开始就使用标准库来写程序,随着讲述的逐渐深入,又一解释这些标准库组件所依赖的基础概念.另外,和其他C++教材不同的是,这本书以实例拉动语言和标准库的讲解,对后两者的讲解是为了给实例程序提供支持,而不是像绝大多数C++教材那样,例子只是用作演示语言特性和标准库用法的辅助工具.

作者在C++领域的编程实践,教育培训以及技术写作方面都是世界一流水准.我喜欢这种大量使用标准库和C++语言原生特性的清新的写作风格.在这本教材面前,几乎迄今为止的所有C++教材都黯然失色或显得过时.尽管这本教材也许对于国内的高校教育来说有些前卫,

55、1. a small C++ program

```
#include <iostream>
```

```
int main()
{
    std::cout<<&quot;Hello, World.&quot;&&lt;&&lt;std::endl;
    return 0;
}
```

(1) The // Characters begin a comment, which extends to the end of the line.

(2) On #include, many c++ facilities are part of standard library rather than being part of core language.

(3) The main function is required to yield a integer as a result. A zero value indicates success, any other value means there was a problem.

2. Some concepts

comment, core language, standard library, #include directives, angle brackets, standard headers, function, statements, curly brackets, output operator, standard output stream, expression, result, side effect, type, left-associative, manipulator, scope, namespace, scope operator, qualified name, free form, string literals, definitions, main function, semicolons, expression statement, null statement.

56、不错的一本入门书，可以说是learning cpp by example的典范

57、没有传说的那么好，起码对我来说，读完了，还是不会用C++，这与我当时没有动手有点关系。但我倒是读了国内的一本大学C++教材和做了大量的C++面试题，才真正的算是会用C++了，另外STL还需要加强一下，这方面的练习做得太少了。

58、唯一从头到尾完整敲过的C++书

59、本来想用最好来形容的，不过自个读过的技术书籍，两双手都能数得过来，就不冒充内行了

。

采用了明显不同于传统教程的路径设计，这本书从一开始就使用标准库来进行讲授，在一个个实例之中穿插进去了基本的知识，如变量，控制流，函数等。作者的编写哲学如同奥卡姆剃刀，“如非必要，勿增实体”。C++的知识林林总总，或者说昏昏晃晃里面都是，充满了各种各样的细节，如何能够将最少的必要知识传递给读者，就看作者本事了。

好的写作者就如同一流的雕塑师，重要的不是留下了什么，而是做了什么选择，去掉了那些次要的知识。这本书的作者做到了此点。和相当部分的教材使用变量--控制流--函数--类--继承多态--库的讲述路径不一样，这本书是本着实用性为原则，尽可能的展现给读者如何使用C++中现有的工具去构建程序，底层的知识固然重要，但是做程序设计的人，尤其是初学者，还是先有个整体的概念为佳。

60、 #include <iostream>

#include <string>

int main()

{

std::cout<<"Your name,please."&<&<std::endl;

std::string name;

std::cin>>name;

std::cout<<"Hello,"&<&<name&<&<"."&<&<std::endl;

return 0;

}

1. Variable is a place to hold values.

2. A variable is an object that has a name. A object is a part of computer's memory that has a type. It is possible to have an object that don't have a name.

3. Three events cause system to flush the buffer: (1) buffer is full. (2) read occurs. (3) explicitly do it (std::endl).

```
//output framing name:
#include <iostream>;
#include <string>;
int main()
{
    std::cout<<<"Your name, please."<<<std::endl;
    std::string name;
    std::cin>>name;
    const std::string greeting = std::string("* ") + "Hello, " + name + "! *";
    int len = greeting.length();
    const std::string decoration(len, '*');
    std::string hallow(len-2, ' ');
    hallow = "*" + hallow + "*";

    std::cout<<<decoration<<<std::endl<<<hallow<<<std::endl<<<greeting<<<std::endl<<<hallow<<<std::endl<<<decoration<<<std::endl;
    return 0;
}
```

Three new ideas:

(1) We can give a variable a value as we define it.

(2) We can use '+' to concatenate a string and a string literal. We can say '+' is overloaded.

(3) We can use const as a part of the definition, indicating that the value of that variable won't change in the rest of its lifetime.

A good property of an operator that never changes is its associativity. '+' is still left-associative.

61、作者在C++方面有丰富的使用经验，所以在语言的讲解方面游刃有余。

这本书通过代码(非玩具代码)展现了C++的一些主要的特征，对初学者来说起到了高屋建瓴的作用，对于有C++基础的人来说，也是一种知识和理解的梳理，本书文笔清晰，内容实在，个人打6星。(ps:强烈建议看英文版，我之前看翻译版的，阅读速度还不如看原版的快)

62、你的这本accelerated c++是机械工业出版社的中文版吗？看完了吗？能不能卖给我？

63、有其他语言基础的人接触C++，看看在C++里是如何处理的，这本书挺合适

64、入门最佳，无奈限于篇幅，不能尽现OO精华

65、看看序言就知道作者也说最好是对C++有一定了解的人对本书最适合。

我也认为。因为我亲身感受了。这本书是我买的第一本C++书籍。

我看呀看就越看越就得书中讲的不知所云。不是书不好，是对于一个新人来说C++的名词太多也较为抽象。不是能很快理解的。而这本书每一章之间的跨的相对较大，加之大量相关内容都没有讲到(书薄吗)所以就不能很好跟上。于是我看到第四章以后实在进展不大。所以该读C++Primer。现读完后再度次作就能领率到此书的真谛，这是无基础的人读不出来的。所以建议不要拿着本书破处。鄙人一家之言，仅供交流。

66、我是看的电子版

67、越来越觉得，技术类书籍序言真的是能忽悠

68、不错

69、倒数第二、三章和附录没有看，等需要的时候再回来细读。

#2015/05/08 重新开始读，这次要踏踏实实地看。

@2015.12.02 重读。

70、搭配C++ Primer入门挺不错

71、我觉得Andrew Koeni的书都很nice，因为他对C++的讲解从来都是站在思想的高度，让读者领

略C++的强大魅力

72、大概是最精简但最适合程序员的C++教程了

精彩书评

- 1、快速的掌握 C++ 的主要知识 最有用的部分以 80% 场景为主 非常适合入门 不被细节所拖累
- 2、这是我通读完的第一本C++教程，之前只读过primer的前几章。不同于其他那些砖一样的入门书籍，Accelerated C++只有短短不到300页，而又是以实例开始逐步深入，对于初学者而言是莫大的福音。然而，要想读透本书，完成所有课后习题，绝非急功近利者短期能够完成的。所以，用深入浅出来形容Accelerated C++再恰当不过了。作为初学者，个人以为此书也并非完美。不知道是我没读透的缘故还是翻译的问题，书中的一些描述（包括少数习题）令人十分费解，相关部分重读多遍依然无法理解作者的含意。个人的看法，如果英语了得，尽量还是读原版。没有习题答案也是件尴尬的事情。做不出来的自不消说，有些题目费劲力气、勉强完成又不知道是否真正达到了题目的要求和作者出题的目的。网上对此多有抱怨。。。作为一个初学者，这个恐怕只有多读两遍再看了。
- 3、<http://www.parkscomputing.com/accelerated-cpp-solutions/> 上面鏈接的答案還沒有寫完，若還需要搜某題的答案，可使用 " Accelerated C++ exercise 3-2 " 這個格式搜。以上是在 <http://www.cnblogs.com/logitechlike/archive/2012/01/04/2312589.html> 上看來的。5.2補充評論中@Rockingpepper君找到了本書全部答案：
：<https://github.com/bitsai/book-exercises/tree/master/Accelerated%20C%2B%2B>
- 4、看看序言就知道作者也说最好是对C++有一定了解的人对本书最适合。我也认为。因为我亲身感受了。这本书是我买的第一本C++书籍。我看呀看就越看越就得书中讲的不知所云。不是书不好，是对于一个新人来说C++的名词太多也较为抽象。不是能很快理解的。而这本书每一章之间的跨的相对较大，加之大量相关内容都没有讲到（书薄吗）所以就不能很好跟上。于是我看到第四章以后实在进展不大。所以该读C++Primer。现读完后再度次作就才能领率到此书的真谛，这是无基础的人读不出来的。所以建议不要拿着本书破处。鄙人一家之言，仅供交流。
- 5、这本书，可能是最好的起点之一，当然，绝对不是学习的终点。我在啃了4遍C++ Primer，钻研了几遍effective，more effective，作出了一些小的项目之后，回头来看这本书，觉得本书的内容绝对够资格称得上初学者的圣经。本书的译者，靳志伟先生，在翻译过程中，确实是下了一番功夫，在中国能多有一些这样的翻译，将是一大幸事。但是，我对机械工业出版社，将毫无保留我的憎恨之情，是的，憎恨，你们的排版，纸张，印刷。。。给这么一本经典书抹了黑。能不能反思一下，你们已经把thinking in C 这么一本经典给翻译毁了，你们还要错多久？
- 6、感觉不如primer看起来条理清楚，可能和机械工业出版社那糟糕的排版有关，都是是大段的密密麻麻的文字块，看着就累。
- 7、终于读完了第一遍，当然懵懵懂懂地，不过真有种醍醐灌顶的感觉。大牛的书写的就是不一样，各种抽象一点点展示给你看，应该怎么做，为什么要这么做讲解得很明白。看完这本书，再看那些先讲C再以C+面向对象方式讲解C++的书，简直就是垃圾。C++完全就是一门全新的语言，OOP以及各种抽象概念才是其精髓。最后要说的是，这本书不太适合初入门来看，讲解的是很详细，但C++太庞杂了，虽然自己有C的基础，也熟悉使用过fortran/matlab，也能看懂Python，看这本书还是比较吃力，各种概念一个接一个来。不过，考虑到C++书要么大部头，要么先讲C，从头培养一个很好的概念/抽象理念还是很重要的，所以如果实在找不到好书，这书做为入门也未尝不可，只是在读之前，一定要告诉自己：坚持读完！之后，你绝对会受益匪浅。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com