

Problem Solving with C++, Eighth Edition

C++ 程序设计

(美) Walter Savitch 著 (第8版)
周 靖 译



PEARSON

清华大学出版社

C++程序设计

(第 8 版)

(美) Walter Savitch 著

周 靖 译

清华大学出版社
北 京

内 容 简 介

本书是最优秀的 C++ 入门教材, 深受师生欢迎。作者结合自己多年的教学经验, 根据教学大纲精心设计并编写了书中内容。与此同时, 本书还采用了很多便于读者巩固所学知识的教学特征, 比如各章开始处的小节总览, 书中随处可见的总结框、编程提示和编程陷阱, 各章结尾处的小结、习题和编程项目等。这些非常适合初学者掌握重要的编程概念。

全书共 18 章, 8 个附录。在讲解 C++ 基础知识之后, 直接引导学生深入函数、I/O 流、类、控制流程、命名空间、数组、字符串、指针和动态数组、递归、模板、指针和链表、派生类、异常以及标准模板库。

Simplified Chinese edition copyright © 2012 PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Problem Solving with C++, Eighth Edition by Walter Savitch

EISBN: 978-0-13-216273-9

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2011-5340

本书封面贴有清华大学出版社和 Pearson Education(培生教育出版集团)激光防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C++ 程序设计/(美)萨维奇(Savitch W.)著; 周靖译. --8 版. --北京: 清华大学出版社, 2012. 2

书名原文: Problem Solving with C++: Eighth Edition

ISBN 978-7-302-27899-3

I. ①C… II. ①萨… ②周… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 008882 号

责任编辑: 文开琪

装帧设计: 杨玉兰

责任校对: 周剑云

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 44.5

字 数: 1281 千字

版 次: 2012 年 2 月第 8 版

印 次: 2012 年 2 月第 1 次印刷

印 数: 1~4000

定 价: 119.00 元



译者序：大道之简

最近在看《罗素回忆录：来自记忆的肖像》，罗素的行文向来字字珠玑，具有很强的感染力，能够引起读者的共鸣，能够给读者很多启发。其中有一篇小文介绍他是如何写作的。他在文中讲到：“我希望用最少量的词便能够把每件事情说得一清二楚。我肯花时间设法找出最简洁的方式把某些事情毫不含糊地表达出来，为此，往往不惜牺牲追求美学上优点的一切企图。”在他 21 岁之前，希望自己的写作风格能够接近于约翰·米尔的风格，因为后者有值得他效仿的句型结构和拓展主题的方式。在经过种种尝试之后，罗素终于醒悟，意识到对华丽词藻和张扬写作风格的模仿会诱发一定程度的虚伪性，认识到所有的模仿都是危险的，从而总结出一些简单的写作准则：

如果可以使用一个简单的词，就永远不要使用一个复杂的词；如果想要做一个包含大量必要条件在内的说明，那么尽量把这些必要条件分解到不同的句子中分别说明。

由此联想到我们的教材，一本优秀的教材，其表述方式和语言应该能够使大家都能明白，而不是充斥着只有少数博学之士才能看得懂的行话或者术语。

在这次翻译并修订 Savitch 老教授的《C++程序设计》的过程中，有颇多这样的感受。真正的大师，是不会一味追求形式化、科学化和精致化，使得专业知识与普通读者渐行渐远，直到彼此之间竖起不可逾越的篱笆。真正优秀的作者，是不会把那些普通读者也能明白的事情说得高深莫测，让大家云里雾里的。Savitch 老先生的这本书让人感受到何为真正的大师，何为真正的优秀作者。这本书从 1995 年首次出版以来，经过十多年的考验，其通俗易懂、妙趣横生、与时俱进的特色，深受广大读者喜爱，被誉为“C++入门经典”，“C++入门教材的‘常青树’”。现在立足于 IT 行业的很多精英和骨干，很多都是在她的熏陶下成长起来的。本书目前已经修订到第 8 版，每次修订，都能增加新的特色，能体现时代的教学特征。

《C++程序设计》之所以畅销不衰，与她的特色是分不开的。

第一，她文风朴实，循序渐进，可读性强(对于一本面向 C++初学者的书来说，这是最起码的要求)，而不像某些作者的书，思维跳跃得很厉害，读起来很费劲。大家都知道，二战期间，丘吉尔的演讲和文章最受欢迎，其中一个重要因素就是他善于用普通老百姓也能懂的浅显语言来阐述自己的观点。Savitch 老先生也是这样行文的。他考虑到广大读者的需求，广泛采用浅显易懂的语言来讲授 C++编程知识。为保持这一特色，我在翻译过程中，也尽量如此。无论原著还是译本，宗旨都是循循善诱地引导一个完全不懂 C++的人在短时间内充分熟悉并掌握 C++编程技术。稍微翻阅几页正文，您就能充分体会到这个特点。

第二，本书完全符合标准。本书的示范程序不仅完全符合最新的 ANSI/ISO C++标准，还遵循行业通行的编程风格，这便于读者在学习本书之后写出任何程序员都可以理解、任何编译器都能通过的程序。

第三个特色体现在本书的组织和结构上。时下流行的一个观点是,学习 C++应该先从类学起。没问题,本书在创作时便充分考虑到了这一部分读者的要求。事实上,您可以按照自己喜欢的任何顺序来阅读各章的内容。这方面的详情可参考前言所提供的“依赖图”。由于这是一本真正的教科书,所以每章都提供了丰富的、重点突出的、非常有趣的自测题和编程项目。

第四,编程实例和编程项目贴近生活。文中 30 多个实例,来源于学生的生活圈,比如,信用卡余额、州收入所得税、购买比萨、温度换算、超市定价系统、回文测试、体重指数等,这些都能让我们体会到编程的乐趣。

第五,第 8 版新增了一些习题和视频讲解。借助于这 60 多个视频讲解,我们可以进一步了解解题思路,牢固掌握基础知识。

一本好书,凝聚着作者很多心血。一本好的译作又何尝不是呢?不仅要仔细揣摩作者的意思,还必须在不误改作者本意的基础上用通俗易懂的文字表达出来。译本如原作,此为“信”;文字通达,令国人一目了然,此为“达”;提炼文字,使之有文学价值,此为“雅”。

计算机科学著作虽然不是文学作品,但随着读者水平的提高,也对“雅”提出了新的要求。人们之所以需要这方面的译本,不仅是为了快速消化和吸收国外的最新技术和观点,还为了满足自己的阅读需求。所以对于译本来说,“信”和“达”固然重要,但“雅”也必不可少。这三者之间的关系是:只有做到“信”,才有可能进一步追求“达”,进而实现“雅”。

为了确保正确性,我在翻译之余把书中的示范程序全部“跑”了一遍。根据我以前的经验,每本书都有这样或那样的错误,有的书错误之多,以至于最终还要出版数十页的勘误表。但是本书的测试结果令人惊叹,它的代码具有很高的正确性,这是一些同类教科书无法媲美的。与此同时,为保证正确性,我在翻译时除了参考网上公开的原书勘误,还和原作者进行了积极而卓有成效的沟通,对原著进行的所有改动都是经过授权的。这一过程有效确保了本书简体中文版的代码质量,使之达到了百分之百的正确率。

关于术语,本书简体中文版采用了业内通行的、国内程序员非常熟悉的词汇。基于翻译的本质,我的工作是为读者无障碍地阅读文档,并积极地跟上作者的思路。假如因为某个或者许多稀奇古怪的词而阻碍了阅读,那就是翻译工作的失败,这不是您我希望见到的。

本书第 4 版在国内出版以来,普遍受到读者的关注和欢迎。读者朋友们普遍反映喜欢此书的写作和翻译风格,喜欢穿插于全书的编程提示与陷阱,更喜欢书中难度不一的自测题和编程项目。有很多读者来信索取编程项目的答案。我也亲自做了部分有挑战性的编程项目,这的确是全书的特色和精华。由于这部分答案仅提供给教师,所以我建议大家发邮件到 coo@netease.com 申请,她们将为您提供帮助。

从第 5 版开始,教材特色更加明显。例如,在各章开篇处,提供了能引起读者关注的小节标题(即当前章的大纲),目的是让读者更清楚地理清脉络。新增一章内容专门介绍 STL(标准模板库)。自第 6 版开始,基本沿用了第 5 版的大纲,修订了部分内容,对章节进行了更合理的调整。各章开始处提供了饶有趣味的引文。这些引文要么出自计算机业界大师,权威、有见地,要么出自名著,巧妙地用双关语把互不关联的两种含义结合起来,从而达到诙谐有趣的效果。

最后，我想感谢原作者 Walter Savitch，是他写出了这样一本极其出色的 C++ 编程教材。感谢翻译过程中所涉及的所有人士，他们是文天山、成荣静、文瑞、苏星兰、文开阳、潘子宇、董健、顾浩云、厉明波、乔艳、王永、姜媛媛、李德武、刘纬地、张辉、代永亮、刘琨、文家焱、梁凯威、肖春雷、靳友英和乔宗彬。同时还要感谢我的乖女儿周子衿，她天真活泼、纯真可爱，现在已经很有思想和创见了，古灵精怪的新鲜想法常常给我很多灵感，我希望能够在来年，给她写一本小小说。

同时，也要感谢各位读者对我的支持和信赖，尤其是本书既往版本的读者，他们的反馈和支持让我感到很温暖，同时也激励和鞭策着我尽心尽力，奉献更好的书。更要感谢对本书提出意见和建议的朋友，例如清华大学毕业生张元章和清华大学在校博士生段菲，他们的“火眼金睛”帮助我发现了此书交稿后因为排版问题引发的新错误，对我提高第 8 版的翻译质量起到了很好的作用。感谢各位，我会尽自己的微薄之力，继续做好《C++ 程序设计》后续版本的翻译和维护工作。

衷心祝愿读者朋友能通过本书，开始愉快而激动人心的程序设计之旅！

周 靖
2012 年春于北京



前言

本书适合 C++ 程序设计和计算机科学入门课程使用。阅读本书不要求读者有任何编程经验，也不要要求掌握除了中学代数之外的其他任何数学知识。

本书前几版的读者请阅读关于第 8 版修订内容的小节，前言的其余内容可略过。新读者请阅读前言的全部内容以把握本书脉络。

第 8 版修订内容

第 8 版采用和第 7 版相同的编程体例。第 7 版全部内容予以保留，但进行了以下改进。

- 第 10 章不再使用流类(比如 `ifstream` 和 `ofstream`)介绍继承概念，而是，使用那一章更简单的银行账户例子来介绍。
- 额外添加了有关作用域、参数传递、随机数生成、动态数组的内容，有几章的幻灯片进行了重新制作。
- 新增 25 个编程项目，合并到每章末尾现有的项目中。
- 本书配套网站添加了 15 个新的视频讲解，总数达 54 个。这些视频讲解辅导学生解题和写程序，有助于巩固对关键编程概念的掌握。如果书中某个主题有对应的视频讲解，就会出现一行特殊的注释(视频讲解：……)。

用过第 7 版的教师可以沿用以前的教案，几乎不需要进行任何改动。

主题可以灵活排序

本书允许教师自由地重新安排教学顺序。为了演示这一灵活性，下面推荐了两种方式排列主题顺序。采用任何一种方式阅读本书，都不会影响学习的连贯性。为了在重新排序时确保这种连贯性，可能需要移动个别小节而不是全章。但是，只有较大的、位置便利的小节才需要移动。为了帮助您根据需要自定义一个教学/阅读顺序，图 P.1 展示了一幅依赖图。另外，每章都有“预备知识”小节，解释了学习那一章的每一节之前需要掌握哪些内容。

重新排序 1：提前学习类

为了有效地设计类，学生需要掌握一些基本的工具，比如控制结构和函数定义。这些基础知识在第 1 章～第 6 章介绍。完成第 6 章的学习后，学生就可以开始编写自己的类了。为了提前学习类的知识，可以像下面这样重新安排各章的顺序。

- **基础知识** 第 1 章、第 2 章、第 3 章、第 4 章、第 5 章和第 6 章。这 6 章全面介绍控制结构、函数定义和基本文件 I/O。第 3 章介绍几种额外的控制结构，如果希望尽早学习类，可以考虑推迟这一章的学习。
- **类和命名空间** 第 10 章、第 11 章的 11.1 节和 11.2 节、第 12 章。这些章节全

面介绍了如何定义类、友元、重载操作符和命名空间。

- **数组、字符串和向量** 第7章和第8章。
- **指针和动态数组** 第9章。
- **类中的数组** 第11章的11.3节和11.4节。
- **继承** 第15章。
- **递归** 第14章(也可以推迟到稍晚的时候学习)。
- **指针和链表** 第13章。

可能还要用到以下各章的部分内容。

- **异常处理** 第16章。
- **模板** 第17章。
- **标准模板库** 第18章。

重新排序 2: 略微延后类的学习

在“重新排序 2”中,将先学完所有控制结构,再学习有关数组的基本知识,之后才开始学习类。虽然对类的接触要比“重新排序 1”晚,但还是比本书的默认顺序略微提前一些。

- **基础知识** 第1章、第2章、第3章、第4章、第5章和第6章。这6章全面介绍了控制结构、函数定义和基本文件 I/O。
- **数组和字符串** 第7章、第8章的8.1节和8.2节。
- **类和命名空间** 第10章、第11章的11.1节、11.2节和第12章。这些章节全面介绍了如何定义类、友元、重载操作符和命名空间。
- **指针和动态数组** 第9章。
- **类中的数组** 第11章的11.3节和11.4节。
- **继承** 第15章。
- **递归** 第14章(也可以推迟到稍晚的时候学习)。
- **向量** 8.3节。
- **指针和链表** 第13章。

可能还要用到以下各章的部分内容。

- **异常处理** 第16章。
- **模板** 第17章。
- **标准模板库** 第18章。

面向学生的易用性

一本书必须按恰当的顺序来讲解恰当的主题,这是最起码的要求。另外,在老师和其他有经验的程序员看来,书中的内容必须清晰而又正确,这是另一个最起码的要求。但是,是不是符合这两项要求的书都是好书呢?答案是否定的。书中的内容必须采取有利于初学者使用的方式来编排。在这本入门教科书中,我尽力让学生觉得清楚和友好。本书以前版本的大量学生反馈证明,这种写作风格确实使内容更清晰,能使学生充分享受到学习的乐趣。

ANSI/ISO C++标准

本书完全兼容于符合最新 ANSI/ISO C++标准的编译器。

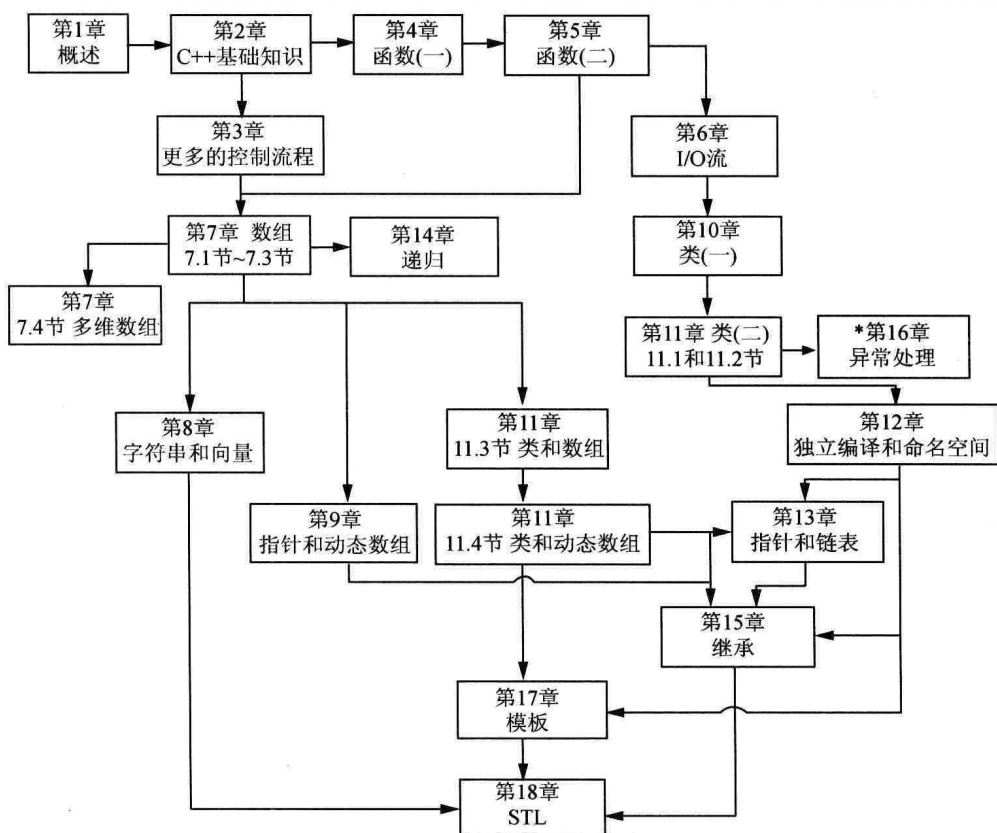
高级主题

许多“高级主题”都成为标准 CS1 课程的一部分。即使它们不是课程的一部分，也最好以补充材料的形式在书中提供。本书提供了大量高级主题，它们既可集成到一门课程中，也可作为自学主题。本书全面讲述了 C++模板、继承(包括虚函数)、异常处理和 STL(Standard Template Library, 标准模板库)。虽然本书使用了库，而且教给学生库的重要性，但不要求任何非标准库。本书只使用所有 C++实现都附带的库。

依赖图

依赖图展示了各个章节可能的排序方式。连接两个框的实线表明上部的框必须先于下部的框完成。只要符合这个条件，采用任何阅读顺序都无损连贯性。如果一个框中包含小节编号，表明该框只代表那些小节，不代表全章。

图 P.1 依赖图



*第16章 偶而会提到派生类，但这些内容是可以忽略的

小结框

每个要点都用一个有底纹的方框来小结。这些“小结框”散布于每一章。

自测题

每章都在重要位置提供大量自测题。每章末尾都有所有自测题的完整答案。

视频讲解

 视频讲解(Video Note)是 Pearson 新推出的可视工具,目的是向学生讲解关键的编程概念和技术。这些短视频演示了从设计到编码来解决问题的过程。视频讲解使学生能方便地自学自己感兴趣的课题,允许选择、播放、倒退、快进和暂停等。每当看到“ 视频讲解:……”,都表明当前主题有一段对应的视频讲解。请自行上网观看视频,网址是 http://media.pearsoncmg.com/aw/aw_savitch_cpp_8/videonotes/vn.html。注意,由于是英文视频,所以为了方便索引,书中保留了这些视频的英文名称。

支持材料

部分支持材料是本书所有读者都能使用的。另一部分仅供有资格的教师使用。

适于本书所有读者的支持材料

- 本书源代码
- PowerPoint 幻灯片
- 视频讲解

为了获取这些材料,请访问本书译者博客(transbot.blog.163.com)。

适于有资格的教师的资源

选用本书作为教材的教师,可致函编辑信箱 wenkaiqi@gmail.com 了解详情。

- 教师资源指南(Instructor's Resource Guide): 包括每一章的教学要点、课堂测验/答案和大量编程项目的答案。
- Test Bank 和 Test Generator: 用于生成试卷。
- PowerPoint 幻灯片: 包括本书的程序和插图。
- Lab Manual(实验手册)。

致 谢

在准备本书素材期间，很多个人和团体提供了大量建议和帮助，并和我进行了广泛而深入的探讨。本书第 1 版大部分内容是我在科罗拉多大学(波德分校)计算机系担任客座教授时写就的。其余内容和后续版本则是在加州大学(圣地亚哥分校)计算机科学与工程系完成的。非常感谢这两所学校为本书写作和教学所提供的良好氛围。

特别感谢向这一版和之前各版提供意见或者编程项目的个人，以下人名按字母排序：

Joseph Allen, Noah Aydin, Claire Bono, Richard Borie, Andrew Burt, Edward Carr, Karla Chaveau, Wei Lian Chen, Joel Cohen, Doug Cosman, Charles Dowling, Scot Drysdale, Joe Faletti, Alex Feldman, Sheila Foster, Paulo Franca, Len Garrett, Jerrold Grossman, Eitan M. Gurari, Andrew Haas, Dennis Heckman, Bob Holloway, Nisar Hundewale, Matt Johnson, Bruce Johnston, Larry Johnson, Thomas Judson, Huzefa Kagdi, Paul J. Kaiser, Michael Keenan, Brian R. King, Paul Kube, Gilliean Lee, Barney MacCabe, Steve Mahaney, Michael Main, Walter A. Manrique, Anne Marchant, John Marsaglia, Nat Martin, Cynthia Martincic, Bob Matthews, Jesse Morehouse, Ethan Munson, Donald Needham, Dung Nguyen, Joseph D. Oldham, Jennifer Perkins, Jeff Roach, Carol Roberts, Ken Rockwood, John Russo, Amber Settle, Naomi Shapiro, Susanne Sherba, Michal Sramka, James Stepleton, Scott Strong, David Teague, Jeffrey Watson, Jerry Weltman, John J. Westman 和 Linda F. Wilson。

还要感谢用过本书前几版的许多老师，他们的建议相当有价值。最后感谢 Kenrick Mock 帮我实现了这一版的全部修订。他的工作令我和编辑激赏！

Walter Savitch

目 录

第 1 章 计算机和 C++编程入门.....1	
1.1 计算机系统.....2	
1.1.1 硬件.....2	
1.1.2 软件.....5	
1.1.3 高级语言.....6	
1.1.4 编译器.....7	
1.1.5 历史回顾.....8	
1.2 编程和问题求解.....9	
1.2.1 算法.....9	
1.2.2 程序设计.....10	
1.2.3 面向对象编程.....12	
1.2.4 软件生命周期.....12	
1.3 C++入门.....13	
1.3.1 C++语言的起源.....13	
1.3.2 一个 C++示范程序.....14	
1.3.3 陷阱：在\n中错误地使用 斜杠.....17	
1.3.4 编程提示：输入和输出语法.....17	
1.3.5 简单 C++程序的布局.....17	
1.3.6 陷阱：在 include 的文件名前 错误地添加一个空格.....18	
1.3.7 编译和运行 C++程序.....19	
1.3.8 编程提示：让程序运行起来.....19	
1.4 测试和调试.....21	
1.4.1 程序错误的分类.....22	
1.4.2 陷阱：错误地假定程序正确.....22	
小结.....23	
自测题答案.....23	
编程项目.....25	
第 2 章 C++基础知识.....27	
2.1 变量和赋值.....28	
2.1.1 变量.....28	
2.1.2 名称：标识符.....30	
2.1.3 变量声明.....31	
2.1.4 赋值语句.....32	
2.1.5 陷阱：未初始化的变量.....33	
2.1.6 编程提示：使用有意义的 名称.....34	
2.2 输入和输出.....35	
2.2.1 使用 cout 进行输出.....35	
2.2.2 include 预编译指令和 命名空间.....36	
2.2.3 转义序列.....37	
2.2.4 编程提示：用\n 或 endl 终止 每一个程序.....38	
2.2.5 格式化带小数点的数字.....39	
2.2.6 用 cin 进行输入.....40	
2.2.7 设计输入和输出.....41	
2.2.8 编程提示：I/O 中的换行.....41	
2.3 数据类型和表达式.....42	
2.3.1 int 类型和 double 类型.....42	
2.3.2 其他数值类型.....43	
2.3.3 char 类型.....44	
2.3.4 bool 类型.....45	
2.3.5 string 类简介.....45	
2.3.6 类型的兼容性.....47	
2.3.7 算术操作符和表达式.....48	
2.3.8 陷阱：除法中的整数.....50	
2.3.9 更多赋值语句.....51	
2.4 简单控制流程.....51	
2.4.1 一个简单的分支机制.....52	
2.4.2 陷阱：连续的不等式.....56	
2.4.3 陷阱：在该用==的时候 错用了=.....56	
2.4.4 复合语句.....57	
2.4.5 简单的循环机制.....58	

2.4.6 递增操作符和递减操作符	61	3.3.7 break 语句	106
2.4.7 编程实例: 信用卡余额	61	3.3.8 陷阱: 嵌套循环中的 break 语句	106
2.4.8 陷阱: 无限循环	63	3.4 设计循环	107
2.5 程序风格	64	3.4.1 求和与求乘积的循环	107
2.5.1 缩进	64	3.4.2 终止循环	108
2.5.2 注释	65	3.4.3 嵌套循环	111
2.5.3 为常量命名	66	3.4.4 调试循环	112
小结	67	小结	114
自测题答案	68	自测题答案	115
编程项目	71	编程项目	118
第3章 更多的控制流程	77	第4章 过程抽象和返回值的函数	123
3.1 使用布尔表达式	78	4.1 自顶向下设计	124
3.1.1 布尔表达式求值	78	4.2 预定义函数	124
3.1.2 陷阱: 将布尔表达式 转换成 int 值	81	4.2.1 使用预定义函数	124
3.1.3 枚举类型(选读)	83	4.2.2 随机数生成	128
3.2 多路分支	83	4.2.3 强制类型转换	129
3.2.1 嵌套语句	83	4.2.4 强制类型转换的古老形式	131
3.2.2 编程提示: 在嵌套语句中 使用花括号	84	4.2.5 陷阱: 整数除法丢弃了 小数部分	131
3.2.3 多路 if-else 语句	86	4.3 程序员自定义函数	132
3.2.4 编程实例: 州收入税	87	4.3.1 函数定义	132
3.2.5 switch 语句	89	4.3.2 能返回布尔值的函数	136
3.2.6 陷阱: 忘记在 switch 语句中 添加 break	92	4.3.3 另一种形式的函数声明	136
3.2.7 为菜单使用 switch 语句	92	4.3.4 陷阱: 实参顺序错误	137
3.2.8 语句块	94	4.3.5 函数定义语法总结	138
3.2.9 陷阱: 疏忽局部变量	95	4.3.6 再论函数定义的位置	139
3.3 C++循环语句详解	97	4.3.7 编程提示: 在分支语句中 使用函数调用	139
3.3.1 while 语句回顾	97	4.4 过程抽象	140
3.3.2 再论递增操作符和递减 操作符	98	4.4.1 黑盒的比喻	140
3.3.3 for 语句	100	4.4.2 编程提示: 选择形参名称	142
3.3.4 陷阱: for 语句中多余的 分号	103	4.4.3 编程提示: 嵌套循环	142
3.3.5 应该使用哪种循环	104	4.4.4 案例分析: 购买比萨	144
3.3.6 陷阱: 未初始化的变量和 无限循环	105	4.4.5 编程提示: 使用伪代码	149
		4.5 局部变量	149
		4.5.1 函数如同小程序	149
		4.5.2 编程实例: 豌豆试验田	151
		4.5.3 全局常量和全局变量	151

4.5.4 传值形参是局部变量.....	153	自测题答案.....	200
4.5.5 块作用域.....	155	编程项目.....	202
4.5.6 再论命名空间.....	156	第 6 章 I/O 流——对象和类入门	207
4.5.7 编程实例：阶乘函数.....	158	6.1 流和基本文件 I/O.....	208
4.6 重载函数名称.....	158	6.1.1 文件之于 I/O 的重要性.....	209
4.6.1 重载入门.....	159	6.1.2 文件 I/O.....	209
4.6.2 编程实例：购买比萨 (修订版).....	161	6.1.3 类与对象入门.....	212
4.6.3 自动类型转换.....	163	6.1.4 编程提示：检查文件是否 成功打开.....	214
小结.....	164	6.1.5 文件 I/O 技术.....	216
自测题答案.....	165	6.1.6 追加到文件(选读).....	218
编程项目.....	167	6.1.7 文件名作为输入(选读).....	219
第 5 章 所有子任务的函数	171	6.2 流 I/O 工具.....	221
5.1 void 函数.....	172	6.2.1 用流函数格式化输出.....	222
5.1.1 void 函数的定义.....	172	6.2.2 操纵元.....	224
5.1.2 编程实例：温度换算.....	174	6.2.3 流作为函数实参.....	226
5.1.3 void 函数中的 return 语句.....	175	6.2.4 编程提示：检查文件尾.....	228
5.2 传引用形参.....	176	6.2.5 命名空间的问题.....	229
5.2.1 初探传引用调用.....	177	6.2.6 编程实例：整理文件格式.....	229
5.2.2 传引用调用详解.....	178	6.3 字符 I/O.....	230
5.2.3 编程实例：swap_values 函数.....	181	6.3.1 get 和 put 成员函数.....	231
5.2.4 混合的参数列表.....	183	6.3.2 putback 成员函数(选读).....	233
5.2.5 编程提示：应该使用 哪种参数.....	183	6.3.3 编程实例：检查输入.....	234
5.2.6 陷阱：疏忽造成的 局部变量.....	184	6.3.4 陷阱：输入中 不期而遇的'\n'.....	235
5.3 使用过程抽象.....	186	6.3.5 编程实例：另一个 new_line 函数.....	236
5.3.1 函数调用函数.....	186	6.3.6 函数的默认实参(选读).....	237
5.3.2 前条件和后条件.....	188	6.3.7 eof 成员函数.....	240
5.3.3 案例分析：超市定价系统.....	188	6.3.8 编程实例：编辑文本文件.....	242
5.4 测试和调试函数.....	192	6.3.9 预定义字符函数.....	243
存根和驱动程序.....	192	6.3.10 陷阱：toupper 和 tolower 返回值.....	245
5.5 常规调试技术.....	196	小结.....	245
5.5.1 不抱成见.....	196	自测题答案.....	246
5.5.2 检查常见错误.....	196	编程项目.....	250
5.5.3 定位错误.....	196	第 7 章 数组	257
5.5.4 assert 宏.....	198	7.1 数组入门.....	258
小结.....	199		

7.1.1	声明和引用数组	258	8.1.4	C 字符串输入和输出	312
7.1.2	编程提示: 为数组使用 for 循环	260	8.1.5	C 字符串到数值的转换和 可靠输入	314
7.1.3	陷阱: 数组索引总是 从零开始	260	8.2	标准 string 类	317
7.1.4	编程提示: 为数组长度使用 一个已定义常量	260	8.2.1	标准类 string 简介	317
7.1.5	数组在内存中的表示	261	8.2.2	string 类的 I/O	319
7.1.6	陷阱: 数组索引越界	262	8.2.3	编程提示: getline 的 其他版本	322
7.1.7	初始化数组	263	8.2.4	陷阱: 混合使用 “cin >> 变量;” 和 getline	322
7.2	函数中的数组	264	8.2.5	用 string 类进行字符串处理	323
7.2.1	索引变量作为函数参数	264	8.2.6	编程实例: 回文测试	326
7.2.2	整个数组作为函数参数	266	8.2.7	string 对象和 C 字符串 之间的转换	328
7.2.3	const 参数修饰符	268	8.3	向量	329
7.2.4	陷阱: 使用 const 参数 修饰符的使用不一致	269	8.3.1	向量基础知识	329
7.2.5	返回数组的函数	270	8.3.2	陷阱: 使用方括号时超出 向量长度	332
7.2.6	案例分析: 产量图	270	8.3.3	编程提示: 向量赋值具有 良好行为	332
7.3	数组编程	280	8.3.4	效率问题	332
7.3.1	部分填充数组	280	小结		333
7.3.2	编程提示: 不要吝啬形参	282	自测题答案		334
7.3.3	编程实例: 搜索数组	282	编程项目		335
7.3.4	编程实例: 数组排序	284	第 9 章 指针和动态数组		341
7.4	多维数组	287	9.1	指针	342
7.4.1	多维数组基础	287	9.1.1	指针变量	343
7.4.2	多维数组参数	288	9.1.2	基本内存管理	348
7.4.3	编程实例: 二维打分程序	289	9.1.3	陷阱: 虚悬指针	348
7.4.4	陷阱: 在数组索引之间使用 逗号	292	9.1.4	静态变量和自动变量	348
小结		292	9.1.5	编程提示: 定义指针类型	349
自测题答案		293	9.2	动态数组	350
编程项目		296	9.2.1	数组变量和指针变量	351
第 8 章 字符串和向量		303	9.2.2	创建和使用动态数组	352
8.1	字符串的数组类型	304	9.2.3	指针运算(选读)	356
8.1.1	C 字符串值和 C 字符串 变量	305	9.2.4	多维动态数组(选读)	357
8.1.2	陷阱: 为 C 字符串 使用=和==	307	小结		358
8.1.3	<cstring>中的其他函数	309	自测题答案		359
			编程项目		359

第 10 章 定义类	363
10.1 结构.....	364
10.1.1 用于异种数据的结构.....	364
10.1.2 陷阱：结构定义中忘记 添加分号.....	367
10.1.3 结构作为函数参数.....	368
10.1.4 编程提示：使用层次化 结构.....	368
10.1.5 对结构进行初始化.....	370
10.2 类.....	372
10.2.1 定义类和成员函数.....	372
10.2.2 公共成员和私有成员.....	375
10.2.3 编程提示：将所有成员 变量设为私有.....	381
10.2.4 编程提示：定义取值函数和 赋值函数.....	381
10.2.5 编程提示：将赋值操作符 用于对象.....	382
10.2.6 编程实例：BankAccount 类 (版本 1).....	382
10.2.7 总结类的一些特征.....	386
10.2.8 用于初始化的构造函数.....	387
10.2.9 编程提示：总是包括默认 构造函数.....	392
10.2.10 陷阱：无参数构造函数.....	393
10.3 抽象数据类型.....	395
10.3.1 用于生成抽象数据 类型的类.....	395
10.3.2 编程实例：类的另一种 实现.....	398
10.4 继承.....	401
10.4.1 派生类.....	401
10.4.2 定义派生类.....	402
小结.....	405
自测题答案.....	405
编程项目.....	409

第 11 章 类中的友元函数、 重载操作符和数组.....413

11.1 友元函数.....	414
----------------	-----

11.1.1 编程实例：一个相等性 函数.....	414
11.1.2 友元函数.....	416
11.1.3 编程提示：定义取值函数和 友元函数.....	418
11.1.4 编程提示：同时使用成员 函数和非成员函数.....	418
11.1.5 编程实例：Money 类 (版本 1).....	419
11.1.6 实现 digit_to_int(选读).....	423
11.1.7 陷阱：数字常量中的 前置零.....	424
11.1.8 const 参数修饰符.....	425
11.1.9 陷阱：修饰符 const 的 用法不一致.....	427
11.2 重载操作符.....	429
11.2.1 重载操作符.....	429
11.2.2 用于自动类型转换的 构造函数.....	432
11.2.3 重载一元操作符.....	433
11.2.4 重载>>和<<.....	434
11.3 数组和类.....	440
11.3.1 类数组.....	440
11.3.2 数组作为类成员.....	443
11.3.3 编程实例：用于部分填充 数组的类.....	444
11.4 类和动态数组.....	445
11.4.1 编程实例：字符串变量类.....	446
11.4.2 析构函数.....	448
11.4.3 陷阱：指针作为传值参数.....	450
11.4.4 拷贝构造函数.....	451
11.4.5 重载赋值操作符.....	454
小结.....	457
自测题答案.....	457
编程项目.....	463

第 12 章 独立编译和命名空间.....469

12.1 独立编译.....	470
----------------	-----

12.1.1 ADT 回顾.....	470	自测题答案	522
12.1.2 案例分析: 独立编译的 DigitalTime 类	471	编程项目	523
12.1.3 使用#ifndef	478	第 14 章 递归	529
12.1.4 编程提示: 定义其他库	480	14.1 面向任务的递归函数	530
12.2 命名空间	481	14.1.1 案例分析: 垂直数字	531
12.2.1 命名空间和 using 预编译 指令	481	14.1.2 深入递归	536
12.2.2 创建命名空间	482	14.1.3 陷阱: 无穷递归	537
12.2.3 限定名称	484	14.1.4 用于递归的栈	538
12.2.4 命名空间的微妙之处 (选读)	485	14.1.5 陷阱: 栈溢出	539
12.2.5 无名命名空间	486	14.1.6 递归与迭代	539
12.2.6 编程提示: 为命名空间 选择名称	489	14.2 面向值的递归函数	540
12.2.7 陷阱: 混淆全局命名空间和 无名命名空间	490	14.2.1 要返回值的递归函数的 常规形式	540
小结	491	14.2.2 编程实例: 另一个求乘方 函数	540
自测题答案	491	14.3 递归思想	544
编程项目	492	14.3.1 递归设计技术	544
第 13 章 指针和链表	495	14.3.2 案例分析: 二叉搜索 (递归思想示例)	545
13.1 节点和链表	496	14.3.3 编程实例: 递归成员函数	550
13.1.1 节点	497	小结	553
13.1.2 链表	500	自测题答案	553
13.1.3 在表头插入节点	501	编程项目	556
13.1.4 陷阱: 丢失节点	503	第 15 章 继承	559
13.1.5 搜索链表	504	15.1 继承基础	560
13.1.6 指针作为迭代器	506	15.1.1 派生类	560
13.1.7 在列表中插入和删除节点	507	15.1.2 派生类中的构造函数	566
13.1.8 陷阱: 为动态数据结构 使用赋值操作符	509	15.1.3 陷阱: 使用来自基类的 私有成员变量	567
13.1.9 链表的变体	510	15.1.4 陷阱: 私有成员函数 不会继承	569
13.1.10 类构成的链表	512	15.1.5 protected 限定符	569
13.2 栈和队列	514	15.1.6 重定义成员函数	571
13.2.1 栈	514	15.1.7 重定义与重载的比较	573
13.2.2 编程实例: 栈类	515	15.1.8 访问重定义的基函数	574
13.2.3 队列	518	15.2 继承细节	575
13.2.4 编程实例: 队列类	518	15.2.1 不继承的函数	575
小结	521		