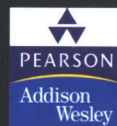


1
国外经典教材



Problem Solving with C++:
The Object of Programming (4th Edition)

C++面向对象程序设计

——基础、数据结构与编程思想(第4版)

(美) Walter Savitch 著
周 靖 译



清华大学出版社

国外经典教材

C++面向对象程序设计

——基础、数据结构与编程思想

(第4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++教材之一, 全球已经拥有数十万读者。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, 作者还以粗体显示了 C++的重要元素, 非常适合初学者掌握重要的编程概念。

全书共 17 章, 9 个附录。在讲解了 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、派生类以及异常。

Simplified Chinese edition copyright © 2003 by **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++: The Object of Programming, 4th Edition

by Walter Savitch, Copyright © 2003

EISBN: 0-321-11347-0

All Rights Reserved.

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

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: 01-2003-0564

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

C++面向对象程序设计——基础、数据结构与编程思想(第4版)/(美)萨维奇(Savitch, W)著;周靖译. —北京:清华大学出版社, 2003.11

(国外经典教材)

书名原文: Problem Solving with C++: The Object of Programming, 4th Edition

ISBN 7-302-07594-8

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

中国版本图书馆 CIP 数据核字(2003)第 103442 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

客户服务: 010-62776969

文稿编辑: 文开棋

封面设计: 久久度企划

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 38.5 字数: 1010 千字

版 次: 2003 年 12 月第 1 版 2003 年 12 月第 1 次印刷

书 号: ISBN 7-302-07594-8/TP·5588

印 数: 1~5000

定 价: 59.00 元

译者序

写一本书很难，翻译一本书更难，翻好一本书更是难上加难。一个好的译本，倾注了译者很多心血，不仅要仔细揣摩作者的意思，还必须用通俗易懂的文字令读者能轻松阅读。在这期间，必须注意不能任意篡改作者的原意。忠实于原著，读译本如读原作，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，由于毕竟不是文学作品，所以即使是原作，也很少有做到“雅”的。人们之所以需要这方面的译本，主要还是为了快速消化和吸收国外的最新技术和观点，所以对于译本来说，最重要的恐怕是“信”和“达”。只有做到“信”，才有可能进一步追求“达”，进而实现“雅”。

《C++面向对象程序设计——基础、数据结构与编程思想》(*Problem Solving with C++: The Object Of Programming*)从1995年首次出版以来，已经培养了很大一批杰出的C++程序员，她的读者遍布全球，在世界各地都有很好的口碑。早在两年前，我在大学里任教的朋友就向我介绍过这本书，说这本书是一本非常好的C++入门教材，尤其适合打算接触C++编程的初学者使用，其编排体系和写作风格也非常适合用作教材，所以特别希望能看到本书的简体中文版。当时，此书还只是第3版。从网上获得的信息来看，这的确是一本精心编写的优秀教材。

后来，有一天得知本书第4版简体中文版已授权清华大学出版社出版，当时非常高兴，立即请命，希望担纲此书的翻译。在众多前来试译的人中，我有幸入围。在拿到样书，并通读全书之后，得出一个结论：这是笔者从事专业计算机图书翻译工作10年时间里，最棒的一本C++入门书。

她之所以好，是因为有三大特色。第一，她文风朴实，循序渐进（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。翻译过程中，我也尽量做到这一点。无论原著还是译本，宗旨都是循循善诱地引导一个对C++完全陌生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，你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，能写出任何程序员都能理解、任何编译器都能通过的程序。

第三，这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，你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。

翻译过程中，为了确保正确性，笔者亲自完成了其中所有的自测题和大部分编程项目。除此之外，还亲自运行了全书所有示范程序。根据笔者的经验，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。但对这本书的测试结果令人惊叹，它的代码具有很高的正确率，这是一些同类教科书无法媲美的。与此同时，为保证正确性，笔者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，这不是你我所希望见到的。

在此，首先要感谢原作者 Walter Savitch，是他写出了这样一本极其出色的 C++ 编程书籍。同时还要感谢翻译过程中为我提供帮助的所有人士，尤其要感谢我的文稿编辑文开棋，是她以极大的耐心审阅了全书，使其无论在质量还是版式上，都上升到了一个新的台阶；还要感谢尤晓东，是他对内容及版式提出中肯了而有价值的意见，并使之付梓出版。

周靖于北京
2003 年 10 月 21 日，大风

前言

本书供 C++ 语言编程和计算机科学的入门课程使用。它不要求读者具有编程经验，而且除了高中代数之外，不需要掌握其他数学知识。

如果你是本书前几版的读者，请仔细阅读后文对第 4 版增补内容的说明，然后可以完全略过前言的剩余部分。如果你第一次接触本书，请仔细阅读前言的剩余部分，其中概述了本书的内容。

第 4 版增补

第 4 版采用的编程体系和第 3 版相同，并采用相同的提纲。对选用本书第 3 版作为教材的老师，可以沿用以前的教案，几乎不需要什么修改。第 4 版的例子进行了更新，添加了更多的习题，另外新增了一些主题。虽然第 3 版也完全兼容于新的 ANSI/ISO 标准，但本版则根据标准进行了更全面的修订。为了反映这一新趋势，我们更新了术语，例如用“函数声明”（Function Declaration）取代了“函数原型”（Function Prototype），用“自由存储”（Freestore）取代了“堆”（Heap）。不过，虽然本书进行了全面更新，但主题的顺序和大多数例子只进行了微小改动。

本书第 4 版是授课老师在课程中增添新主题的好机会。本版增补了向量（vector）的主题。另外还进行了大量不是很明显的增补，比如指针作为迭代器使用，以及动态多维数组的问题等。

自己确定主题顺序

以 C++ 为主题的大多数入门教材都有一个非常详细的进度表，老师必须在上课时严格遵循。但本书不准备这样做。如果你是一名老师，这本书能适应你的教学方式，而不是要求你去适应这本书。你可轻松地更改章节顺序，同时不会失去阅读本书时的连续性。前言末尾的图 P.1 为书中内容展示了一幅依赖图。另外，每一章都包含一个“预备知识”小节，它解释了在学习那一章的内容之前，必须先学习哪些章的内容。这样一来，老师就可以轻松地重新安排各个章节的授课顺序。

虽然本书使用了库，并讲解了库的重要性，但它不要求任何非标准库。本书只使用几乎所有 C++ 实现都提供了的库。

ANSI/ISO C++ 标准

本版进行了全面更新，使用与新的 ANSI/ISO C++ 标准相符的编译器。由于没有任何理由再去使用与新标准不符的编译器，所以本书不再深入讨论如何处理旧版编译器。

提前介绍类

本书允许灵活地讲授类。学生可或早或晚地学习类的知识。

一本书至少可采取 2 种方式来讲述类：提前教学生如何设计他们自己的类，或者只是提前教他们使用类，而不去定义它们。本书提前教学生定义自己的类，而不只是提前教他们使用类。为了有效地设计类，学生需要一些基本工具，比如一些简单的控制结构和函数定义。所以，本书将在第 2 章、第 3 章和第 4 章开始讲解这些基础知识。然后就快速转向类的主题。第 5 章将用文件 I/O 流来教学生使用类。第 6 章将解释如何编写自己的类。

本书采用定量方式来传授类的知识。它先教学生写一些非常简单的类，然后添加构造函数，再重载简单操作符，再重载 I/O 操作符<<和>>，依此类推。这种定量教学方式避免了学生一开始就接触大量复杂的构造和概念。但是，本书的一个目标是让学生尽可能提前地编写现实的类定义，而不是强迫他们花大量时间去编写一些故意简单化的类。到第 8 章结束时，学生就能写出完整和实用的类，实现“类”的课程目标。

在课堂测验中，我们发现学生如果提前接触类的知识，他们的反应和提前接触函数时的反应是一样的。这肯定会带来一些问题。但是，学生完全有能力或早或晚地学习这些知识。此外，通过提前学习类，学生可对类有更好的理解。对于自己掌握的第一种问题求解技术，学生们往往具有很高的忠诚度。所以，作为老师，如果你希望他们真正去使用一种技术，就得提前教它。另外，类并不是学生们遇到的最困难的主题。例如，相较于普通（C 风格）的数组，类显得更直观，行为也更易理解。普通数组的问题将推迟到后文讲述。

虽然我们主张提前讲解类的知识，但我们也知道并不是每个人都想提前讲解类。所以，本书也允许老师推迟讲解类的知识。后面的“主题可以灵活排序”小节将具体讨论这个问题。

继承问题主要在第 5 章讲述，使学生能提前接触这一概念。但是，本书要推迟到后文教学生写自己的派生类，因为在一门初学课程刚开始的时候，继承和派生类的重要性往往显示不出来。第 16 章教学生如何定义和使用派生类，其中包括使用虚函数。有的老师选择在第二门课中再讲述这方面的主题，另一些老师则可能想在入门课中讨论继承问题。如果愿意，有关继承的主题也可提前讲述。

面向学生的易用性

一本书仅按合适的顺序来讲解合适的主题是不够的。除此之外，一本书也不能只是做到清晰和正确，让老师或其他有经验的程序员读起来满意。素材必须采用一种初学者容易使用的方式来编排。在这个系列的入门教材中，我尽力让学生觉得清楚和友好。根据使用本书以前版本的许多学生的反馈，证明这种写作风格确实使素材更清晰，而且通常都能为学生带来乐趣。

高级主题

许多“高级主题”都成为标准的 CSI（计算机科学系一年级）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随书提供。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。我们全面讲述了 C++ 模板、继承（包括虚函数）和异常处理。

总结框

每个要点都用一个有底纹的方框来总结。这些总结框散布于每一章。

自测题

每章都在相应的重要位置提供大量自测题。每章末尾则给出所有自测题的完整答案。

课堂实测

世界各地的许多学生都用过本书的前 3 版。许多学生以及他们的老师都提供了有益的反馈意见，让我了解哪些适用于他们，哪些则不适用。大多数意见都积极地肯定了这本书，表明大多数学生和老师都非常喜欢这本书。当然，还有一些建议要求我对本书进行修订。所有修订意见都已经过仔细考虑。正是基于这些宝贵的意见，这一版才能逐渐成形，最终摆上你的案头。相较于以前各版，这一版能更好地满足学生和老师的需要。

主题可以灵活排序

本书允许老师自由地重新安排教学材料。为了展示这一灵活性，我们推荐了多种方式来排列主题顺序。采取推荐的任何一种方式来阅读本书，都不失连贯性。为了确保这种连贯性，你可能需要移动节而不是整章。但是，只有处于方便位置的大节才需要移动。为了帮助你根据需要来自定义一个阅读顺序，图 P.1 展示了一幅依赖图，每一章都有一个“预备知识”小节，它解释了在继续本章的学习之前，事先需要阅读哪些章。

重新排序 1：推迟学习类

第 4 版基本要求首先完成一门传统 ANSI C 课程的全部学习，然后才学习类的知识。在讲解类之前，惟一和 C++ 相似的就是流 I/O 的使用。由于流 I/O 确实要求用到一些命名空间和类库，所以在前几章里，稍微讲解了怎样使用预定义命名空间和一些标准库 I/O 类。

- 基础知识：第 1 章、第 2 章、第 3 章、第 4 章、第 5 章和第 7 章（省略涉及到类定义的第 6 章）。这几章覆盖了所有控制结构、函数定义以及基本文件 I/O 的知识。
- 数组：第 10 章，省略涉及类的使用的 10.4 节。

- C 风格的字符串：第 11 章的 11.1 节。
- 指针和动态数组：第 12 章，省略和类有关的最后一节（12.3 节）。
- 递归：第 13 章，省略本章结尾的有关类的编程例子（另外，也可将递归移动到课程后期讲述）。
- 结构和类：第 6 章、第 8 章和第 11 章的 11.2 节。
- 独立编译和命名空间：第 9 章。
- 指针和链表：第 15 章。

也可使用以下章节的任意组合：

- 向量：第 11 章的 11.3 节。
- 模板：第 14 章。
- 继承：第 16 章。
- 异常处理：第 17 章。

重新排序 2：提前学习类

这个版本覆盖所有控制结构和有关数组的基本知识，然后才开始讲解类。只是讲解类的时机要比上一种排序方式稍微提前。

- 基础知识：第 1 章、第 2 章、第 3 章、第 4 章、第 5 章和第 7 章（省略涉及到类定义的第 6 章）。这几章覆盖了所有控制结构、函数定义以及基本文件 I/O。
- 数组：第 10 章，省略涉及类的使用的 10.4 节。
- C 风格的字符串：第 11 章的 11.1 节。
- 结构和类：第 6 章、第 8 章和第 11 章的 11.2 节。
- 独立编译和命名空间：第 9 章。
- 指针和动态数组：第 12 章。
- 递归：第 13 章。
- 指针和链表：第 15 章。

也可使用以下章节的任意组合：

- 向量：第 11 章的 11.3 节。
- 模板：第 14 章。
- 继承：第 16 章。
- 异常处理：第 17 章。

重新排序 3：提前学习类，但在此之前先掌握所有控制结构

这几乎是本书最标准的排序方式。你只需将第 7 章放到第 6 章之前。

支持材料

原出版商在以下网站提供了以下支持材料：

<http://www.aw.com/savitch/>

- 示范程序：本书所有程序都可在上述网站找到。
- PowerPoint 幻灯片：包括程序和本书插图。

另外，也可到作者的网站下载程序：<http://www-cse.ucsd.edu/users/savitch/>

以下材料只提供给通过认证的教师。欲知详请，请联系 Addison-Wesley 销售代表。

- 教参：每章都有教参，包括丰富的教学提示、课堂测验和答案以及许多编程项目的答案。
- Test Bank：一个强大的试卷生成软件。
- Lab Manual（上机实验手册）。

电子邮件联系

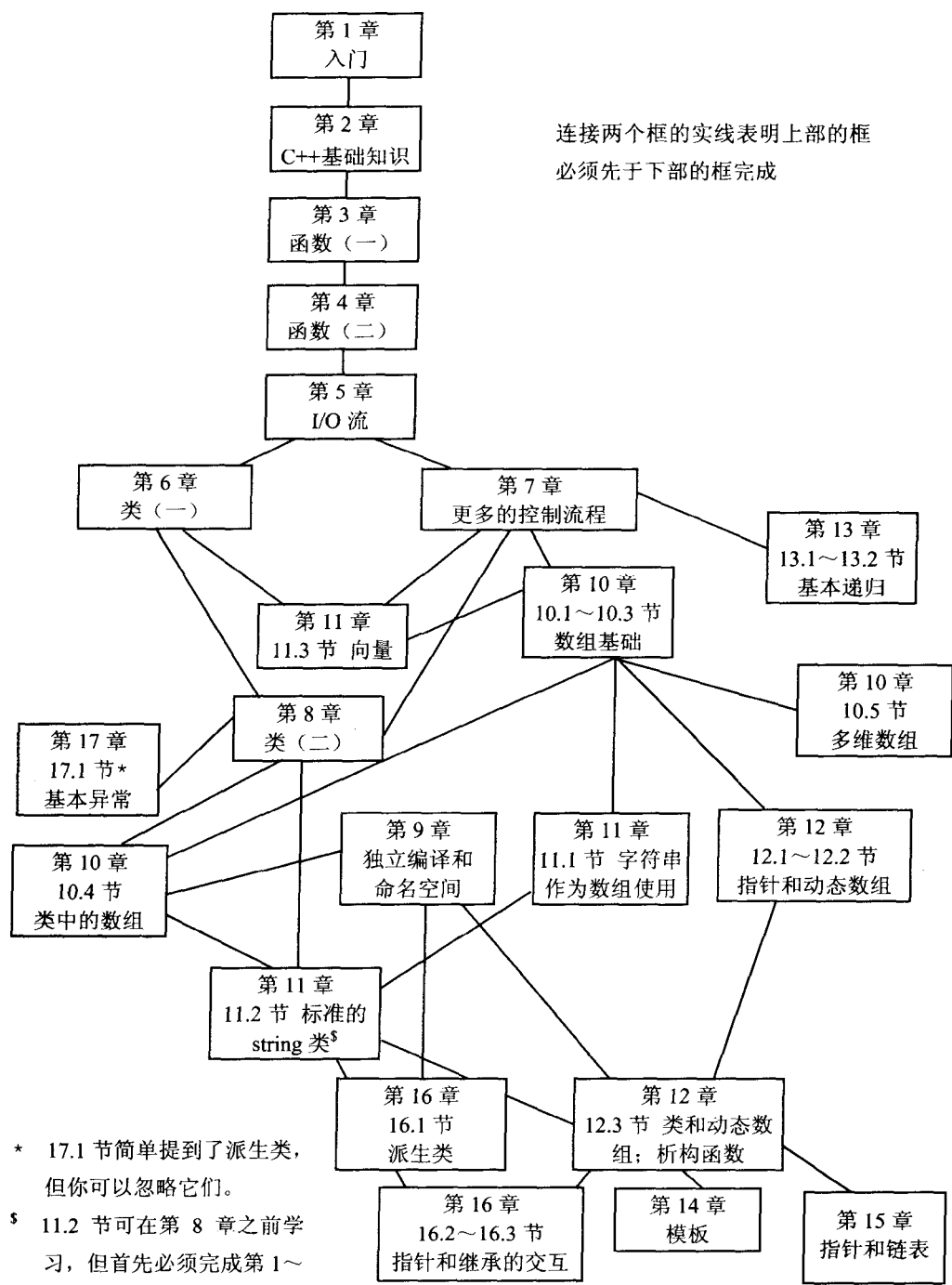
非常欢迎你的宝贵意见和建议，这样能继续改进本书，使之更符合你的需要。所有意见和建议请发到：

wsavitch@ucsd.edu

我想知道你对这本书的任何看法，并希望你提出修订建议。但遗憾的是，我不能为学生提供电子邮件咨询或者辅导服务，否则我的电子邮件会堆积如山。尤其是，我不能为本书的习题或者你的老师布置的其他作业提供答案，因为我没有足够的时间来解答所有这些问题。另外，我也不希望干扰任何老师的教学计划。他们对于学生应该如何解决编程问题，可能会有自己的一些安排。如果你确实需要帮助，那么至少值得安慰的是，本书已经包括了所有自测题的答案。另外在教参中，还为教师提供了部分“编程项目”的答案，但它们只提供给用本书来教学的老师，不可泄露给学生。

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框中的内容必须先于下部框中的内容进行学习。只要符合这个条件，那么采用任何阅读顺序都不会损失连贯性。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。



* 17.1 节简单提到了派生类，但你可以忽略它们。

§ 11.2 节可在第 8 章之前学习，但首先必须完成第 1~7 章的学习。

图 P.1 依赖图

简明目录

前言	iii
第 1 章 计算机和 C++编程入门	1
第 2 章 C++基础知识	26
第 3 章 过程抽象和返回一个值的函数	72
第 4 章 所有子任务的函数	112
第 5 章 I/O 流——对象和类入门	142
第 6 章 定义类	194
第 7 章 更多的控制流程	239
第 8 章 友元函数和重载操作符	287
第 9 章 独立编译和命名空间	321
第 10 章 数组	345
第 11 章 字符串和向量	397
第 12 章 指针和动态数组	430
第 13 章 递归	462
第 14 章 模板	490
第 15 章 指针和链表	509
第 16 章 继承	532
第 17 章 异常处理	566
附录 1 C++关键字	586
附录 2 操作符的优先级	587
附录 3 ASCII 字符集	588
附录 4 部分库函数	589
附录 5 assert 语句	592
附录 6 内联函数	593
附录 7 重载数据索引方括号	594
附录 8 this 指针	595
附录 9 将操作符重载为成员操作符	597

详细目录

第1章 计算机和C++编程入门.....1	
概述.....1	
1.1 计算机系统.....1	
1.1.1 硬件.....1	
1.1.2 软件.....5	
1.1.3 高级语言.....6	
1.1.4 编译器.....6	
自测题.....8	
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.....11	
1.2.3 面向对象编程.....12	
1.2.4 软件生存期.....12	
自测题.....13	
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中使用错误的斜杠.....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文件名前 添加一个空格.....19	
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.....20	
自测题.....21	
1.4 测试和调试.....21	
1.4.1 程序错误的种类.....22	
1.4.2 陷阱：错误地假定程序正确.....22	
自测题.....23	
小结.....23	
自测题答案.....24	
编程项目.....25	
第2章 C++基础知识.....26	
概述.....26	
预备知识.....26	
2.1 变量和赋值.....26	
2.1.1 变量.....26	
2.1.2 名称：标识符.....28	
2.1.3 变量声明.....29	
2.1.4 赋值语句.....30	
2.1.5 陷阱：未初始化的变量.....31	
2.1.6 编程提示：使用有意义的名称.....32	
自测题.....33	
2.2 输入和输出.....33	
2.2.1 使用cout进行输出.....33	
2.2.2 include预编译指令和命名空间.....34	
2.2.3 转义序列.....36	
2.2.4 编程提示：用\n或endl终止每一个程序.....37	
2.2.5 格式化带小数点的数字.....37	
2.2.6 用cin进行输入.....38	
2.2.7 设计输入和输出.....39	
2.2.8 编程提示：I/O中的行中断.....40	
自测题.....40	
2.3 数据类型和表达式.....41	
2.3.1 int类型 and double类型.....41	
2.3.2 其他数字类型.....42	
2.3.3 char类型.....43	
2.3.4 bool类型.....44	
2.3.5 类型的兼容性.....44	
2.3.6 算术操作符和表达式.....46	
2.3.7 陷阱：除法中的整数.....48	
自测题.....48	
2.3.8 更多赋值语句.....49	
2.4 简单控制流程.....49	
2.4.1 一个简单的分支机制.....49	
2.4.2 陷阱：连续的不等式.....54	
2.4.3 陷阱：错用=来取代==.....54	
2.4.4 复合语句.....55	
自测题.....55	
2.4.5 简单的循环机制.....56	
2.4.6 递增操作符和递减操作符.....60	
2.4.7 编程示例：信用卡余额.....60	
2.4.8 陷阱：无限循环.....61	
自测题.....62	
2.5 程序风格.....63	
2.5.1 缩进.....63	
2.5.2 注释.....63	
2.5.3 为常量命名.....65	
自测题.....66	
小结.....66	
自测题答案.....67	
编程项目.....70	

第3章 过程抽象和返回一个值的函数	72
概述	72
预备知识	72
3.1 自顶向下设计	72
3.2 预定义函数	72
3.2.1 使用预定义函数	73
3.2.2 强制类型转换	76
3.2.3 强制类型转换的古老形式	78
3.2.4 陷阱：整数除法丢弃了小数部分	78
自测题	79
3.3 程序员自定义函数	79
3.3.1 函数定义	79
3.3.2 另一种形式的函数声明	83
3.3.3 陷阱：实参顺序错误	83
3.3.4 函数定义语法总结	85
3.3.5 再论函数定义的位置	85
自测题	86
3.4 过程抽象	86
3.4.1 黑盒的比喻	86
3.4.2 编程提示：选择形参名称	88
3.4.3 案例分析：购买比萨	89
3.4.4 编程提示：使用伪代码	93
自测题	94
3.5 局部变量	94
3.5.1 函数如同小程序	94
3.5.2 编程实例：豌豆试验田	96
3.5.3 全局常量和全局变量	96
3.5.4 传值调用形参是局部变量	98
3.5.5 再论命名空间	99
自测题	101
3.5.6 编程实例：阶乘函数	101
3.6 重载函数名称	102
3.6.1 重载入门	102
3.6.2 编程实例：购买比萨（修订版）	104
3.6.3 自动类型转换	106
自测题	107
小结	107
自测题答案	108
编程项目	110
第4章 所有子任务的函数	112
概述	112
预备知识	112
4.1 void 函数	112
4.1.1 void 函数的定义	112
4.1.2 编程实例：温度换算	114
4.1.3 void 函数中的 return 语句	115
自测题	116
4.2 传引用调用形参	117
4.2.1 初探传引用调用	117
4.2.2 传引用调用详解	119
4.2.3 编程实例：swap_values 函数	121
4.2.4 混合的参数列表	122
4.2.5 编程提示：应该使用哪种参数	123
4.2.6 陷阱：疏忽的局部变量	124
自测题	125
4.3 使用过程抽象	126
4.3.1 由函数来调用函数	126
4.3.2 前条件和后条件	128
4.3.3 案例分析：超市定价系统	128
自测题	132
4.4 测试和调试函数	132
stub 和驱动程序	132
自测题	135
小结	136
自测题答案	136
编程项目	139
第5章 I/O 流——对象和类入门	142
概述	142
预备知识	142
5.1 流和基本文件 I/O	142
5.1.1 文件之于 I/O 的重要性	143
5.1.2 文件 I/O	143
5.1.3 类与对象入门	146
5.1.4 编程提示：检查文件是否成功打开	148
5.1.5 文件 I/O 技术	150
自测题	152
5.1.6 追加到文件（选读）	152
5.1.7 文件名作为输入（选读）	154
5.2 流 I/O 工具	156
5.2.1 用流函数格式化输出	156
5.2.2 操纵元	159
自测题	160
5.2.3 流作为函数实参	161
5.2.4 编程提示：检查文件尾	163
5.2.5 命名空间的问题	164
5.2.6 编程实例：整理文件格式	165
自测题	165
5.3 字符 I/O	165
5.3.1 get 和 put 成员函数	166
5.3.2 putback 成员函数（选读）	168

5.3.3 编程实例：检查输入	169	6.3.2 编程实例：类的另一种实现	230
5.3.4 陷阱：输入中不期而遇的'\n'	171	自测题	232
自测题	172	小结	233
5.3.5 eof 成员函数	173	自测题答案	233
自测题	175	编程项目	237
5.3.6 编程实例：编辑文本文件	175	第7章 更多的控制流程	239
5.3.7 预定义的字符函数	176	概述	239
5.3.8 陷阱：toupper 和 tolower 返回 int 值	178	预备知识	239
自测题	178	7.1 使用布尔表达式	239
5.4 继承	179	7.1.1 布尔表达式求值	239
5.4.1 流类之间的继承关系	179	7.1.2 陷阱：将布尔表达式转换成 int 值	242
5.4.2 编程实例：另一个 new_line 函数	182	自测题	243
5.4.3 函数的默认参数（选读）	183	7.1.3 能返回布尔值的函数	244
自测题	184	自测题	244
小结	185	7.1.4 枚举类型（选读）	245
自测题答案	185	7.2 多路分支	245
编程项目	190	7.2.1 嵌套语句	246
第6章 定义类	194	7.2.2 编程提示：在嵌套语句中使用花括号	246
概述	194	7.2.3 多路 if-else 语句	248
预备知识	194	7.2.4 编程实例：州税	250
6.1 结构	194	自测题	251
6.1.1 用于异种数据的结构	194	7.2.5 switch 语句	252
6.1.2 陷阱：结构定义中忘记一个分号	198	7.2.6 陷阱：忘记在 switch 语句中添加 break	255
6.1.3 结构作为函数参数	198	7.2.7 为菜单使用 switch 语句	255
6.1.4 编程提示：使用层次化结构	199	7.2.8 编程提示：在分支语句中使用 函数调用	256
6.1.5 对结构进行初始化	200	7.2.9 块	257
自测题	201	7.2.10 陷阱：疏忽局部变量	259
6.2 类	202	自测题	259
6.2.1 定义类和成员函数	202	7.3 C++循环语句详解	260
自测题	206	7.3.1 while 语句回顾	260
6.2.2 公共成员和私有成员	206	7.3.2 再论递增操作符和递减操作符	261
6.2.3 编程提示：将所有成员变量变成私有	212	自测题	263
6.2.4 编程提示：定义取值函数和赋值函数	212	7.3.3 for 语句	263
自测题	212	7.3.4 陷阱：for 语句中多余的分号	267
6.2.5 编程提示：将赋值操作符用于对象	213	7.3.5 应该使用哪种循环	268
6.2.6 编程实例：BankAccount 类 (版本 1)	213	自测题	268
6.2.7 总结类的一些特征	217	7.3.6 陷阱：未初始化的变量和无限循环	269
自测题	218	7.3.7 break 语句	269
6.2.8 用于初始化的构造函数	218	7.3.8 陷阱：嵌套循环中的 break 语句	270
6.2.9 编程提示：总是包括一个默认 构造函数	224	自测题	270
6.2.10 陷阱：无参数构造函数	225	7.4 设计循环	271
自测题	226	7.4.1 求和与求乘积的循环	271
6.3 抽象数据类型	226	7.4.2 终止循环	272
6.3.1 用于生成抽象数据类型的类	227	7.4.3 嵌套循环	275

自测题	278	9.2.1 命名空间和 using 预编译指令	332
7.4.4 调试循环	278	9.2.2 创建命名空间	333
自测题	280	自测题	335
小结	280	9.2.3 限定名称	335
自测题答案	281	9.2.4 命名空间的微妙之处 (选读)	336
编程项目	284	自测题	337
第 8 章 友元函数和重载操作符	287	9.2.5 无名命名空间	337
概述	287	9.2.6 编程提示: 为命名空间选择一个名称	341
预备知识	287	9.2.7 陷阱: 混淆全局命名空间和 无名命名空间	341
8.1 友元函数	287	自测题	341
8.1.1 编程实例: 一个相等性函数	287	小结	342
自测题	289	自测题答案	343
8.1.2 友元函数	289	编程项目	344
8.1.3 编程提示: 定义取值函数和友元函数	291		
8.1.4 编程提示: 同时使用成员函数 和非成员函数	291	第 10 章 数组	345
8.1.5 编程实例: Money 类 (版本 1)	293	概述	345
8.1.6 实现 digit_to_int (选读)	297	预备知识	345
8.1.7 陷阱: 数字常量中的前置零	298	10.1 数组入门	345
自测题	298	10.1.1 声明和引用数组	345
8.1.8 const 参数修饰符	299	10.1.2 编程提示: 为数组使用 for 循环	347
8.1.9 陷阱: 不一致地使用 const	300	10.1.3 陷阱: 数组索引总是从零开始	347
自测题	302	10.1.4 编程提示: 为数组长度使用一个 已定义常量	347
8.2 重载操作符	303	10.1.5 数组在内存中的表示	348
8.2.1 重载操作符	303	10.1.6 陷阱: 数组索引越界	349
自测题	306	10.1.7 初始化数组	350
8.2.2 用于自动类型转换的构造函数	306	自测题	350
自测题	307	10.2 函数中的数组	352
8.2.3 重载一元操作符	307	10.2.1 索引变量作为函数参数	352
8.2.4 重载 >> 和 <<	308	自测题	353
自测题	314	10.2.2 整个数组作为函数参数	353
小结	315	10.2.3 const 参数修饰符	356
自测题答案	315	10.2.4 陷阱: 不一致地使用 const 参数	357
编程项目	319	10.2.5 返回数组的函数	358
第 9 章 独立编译和命名空间	321	10.2.6 案例分析: 产量图	358
概述	321	自测题	367
预备知识	321	10.3 数组编程	367
9.1 独立编译	321	10.3.1 部分填充数组	367
9.1.1 ADT 回顾	321	10.3.2 编程提示: 不要吝啬形参	369
9.1.2 案例分析: 一个独立编译的 类 DigitalTime	322	10.3.3 编程实例: 数组搜索	370
9.1.3 使用 #ifndef	329	10.3.4 编程实例: 数组排序	371
9.1.4 编程提示: 定义其他库	331	自测题	374
自测题	331	10.4 数组和类	375
9.2 命名空间	332	10.4.1 类数组	375
		自测题	378

10.4.2 数组作为类成员	378
10.4.3 编程实例：用于部分填充数组的 一个类	379
自测题	381
10.5 多维数组	381
10.5.1 多维数组基础	381
10.5.2 多维数组参数	382
10.5.3 编程实例：二维打分程序	383
10.5.4 陷阱：在数组索引之间使用逗号	386
自测题	386
小结	387
自测题答案	387
编程项目	391
第 11 章 字符串和向量	397
概述	397
预备知识	397
11.1 字符串的一种数组类型	397
11.1.1 C 字符串值和 C 字符串变量	398
11.1.2 陷阱：为 C 字符串使用=和==	400
11.1.3 <cstring>中的其他函数	402
自测题	403
11.1.4 C 字符串输入和输出	404
自测题	406
11.1.5 C 字符串到数字的转换和可靠输入	406
11.2 标准 string 类	410
11.2.1 标准类 string 简介	410
11.2.2 string 类的 I/O	412
自测题	414
11.2.3 编程提示：getline 的其他版本	414
11.2.4 陷阱：混合使用“cin >> 变量,” 和 getline	415
11.2.5 用 string 类进行字符串处理	416
11.2.6 编程实例：回文测试	418
自测题	421
11.2.7 string 对象和 C 字符串之间的转换	421
11.3 向量	422
11.3.1 向量基础知识	422
11.3.2 陷阱：使用方括号时超出向量长度	424
11.3.3 编程提示：向量赋值具有良好行为	424
11.3.4 效率问题	425
自测题	426
小结	426
自测题答案	426
编程项目	428
第 12 章 指针和动态数组	430
概述	430
预备知识	430
12.1 指针	430
12.1.1 指针变量	431
自测题	436
12.1.2 基本内存管理	436
12.1.3 陷阱：虚悬指针	437
12.1.4 静态变量和自动变量	437
12.1.5 编程提示：定义指针类型	438
自测题	439
12.2 动态数组	439
12.2.1 数组变量和指针变量	439
12.2.2 创建和使用动态数组	440
自测题	443
12.2.3 指针运算（选读）	444
自测题	445
12.2.4 多维动态数组（选读）	445
12.3 类和动态数组	446
12.3.1 编程实例：一个字符串变量类	447
12.3.2 析构函数	450
12.3.3 陷阱：指针作为传值调用参数	451
12.3.4 拷贝构造函数	452
自测题	455
12.3.5 重载赋值操作符	456
自测题	458
小结	458
自测题答案	458
编程项目	460
第 13 章 递归	462
概述	462
预备知识	462
13.1 面向任务的递归函数	462
13.1.1 案例分析：垂直数字	462
13.1.2 深入递归	467
13.1.3 陷阱：无穷递归	468
自测题	468
13.1.4 用于递归的堆栈	469
13.1.5 陷阱：堆栈溢出	470
13.1.6 递归与迭代	470
自测题	471
13.2 面向值的递归函数	471
13.2.1 返回值的递归函数的泛型	471
13.2.2 编程实例：另一个 Powers 函数	472
自测题	474