- 畅销数十年的C++入门经典教材
- 结构严谨、清晰，行文诙谐幽默，妙趣横生
- 80+技巧提示，30+生动有趣的编程实例
- 60+精彩视频，410+自测题，220+编程项目

C++ 面向对象程序设计

（第7版 影印版）

（美）Walter Savitch 著

清华大学出版社

C++ 面向对象程序设计

（第7版 影印版）（美）Walter Savitch 著

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，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。

图书在版编目(CIP)数据
C++面向对象程序设计(第 7 版 影印版)=Problem Solving with C++，Seventh Edition：第 7 版：英文/(美)萨维奇(Savitch W.)著. --影印本. --北京：清华大学出版社，2011.1
ISBN 978-7-302-24546-9
Ⅰ. ①C…　Ⅱ. ①萨…　Ⅲ. ①C 语言—程序设计—英文　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 245436 号

责任编辑：文开琪
责任校对：周剑云
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　销：全国新华书店
开　本：185×230　**印　张**：67.25　**字　数**：1002 千字
版　次：2011 年 1 月第 1 版　**印　次**：2011 年 1 月第1 次印刷
印　数：1～3500
定　价：98.00 元

产品编号：040737-01

Preface(中文版)

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

本书前几版的读者，请仔细阅读以下解释第 7 版修订内容的小节，前言的其余内容则可以跳过不看。但本书的新读者，务必仔细阅读前言的所有内容，以便把握本书的脉络。

第 7 版修订内容

第 7 版采用和第 6 版相同的编写体例，并保留了第 6 版的所有内容。新增了 30 个编程项目，并对几章原有的编程项目进行了改编。额外增加了有关 map 类的说明，而且示例程序进行了更新，以便同当今使用的大多数编译器兼容。此外，本书配套网站增加了 39 段视频，讨论了特定的主题以及编程项目的解决方案。这些视频可辅导学生完成解决问题和编写程序的过程，并有助于加强对关键编程概念的理解。如果书中的某个主题有对应的视频，就会出现一行特殊的注释(Video Note：……)。最后，根据授课教师的反馈，有关继承的介绍性材料从第 6 章“I/O 流”移到了第 10 章“定义类”。然而，对于用过第 6 版的授课教师，可以继续沿用以前的教案，几乎不需要进行任何改动。

自主决定主题顺序

C++主题的大多数入门教科书都有一个非常详细的进度表，授课教师必须严格遵循。但本书不准备这样做。本书能适应授课教师的教学方式，而不是要求教师来适应本书。教师可以轻松地调整章节顺序，而不失连贯性。前言末尾的图 P.1 展示了本书所有主题的一幅“依赖图”，它描述了各个章节的多种教学顺序。另外，每章都包含一个“预备知识”小节，说明在学习当前章内容之前，必须先学习本书哪些部分的内容。这样一来，授课教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，并强调了库的重要性，但不要求任何非标准的库。本书只使用几乎所有 C++实现都会提供的库。

类的介绍可灵活提前或推后

本书允许灵活地讲授类。授课教师可以根据需要提前或推迟讲类的主题。

本书的默认安排顺序是：首先向学生讲述变量声明、表达式求值、控制结构、过程抽象、函数定义、数组和指针处理的基本概念。这样一来，学生能牢固掌握编程概念，为后期掌握面向对象编程所涉及的复杂概念奠定坚实的基础。本书第 2 章～第 9 章介绍最基础的编程概念。第 6 章将用文件 I/O 流教学生如何使用类。由于流的 I/O 要求使用命名空间和类库，因此在前几章，还要介绍如何使用预定义的命名空间和由标准库提供的 I/O 类。在第 10 章，学生们将学习如何编写自己的类。

打算在课程前期就介绍类的授课教师，则可以重新安排各章的顺序以便适应自己的教学需求。这方面的细节将在下页的“主题可以灵活排序”一节介绍。一般而言，授课教师可以在完成第 6 章的教学后立即开始第 10 章“定义类”。

本书采用定量方式来传授类的知识。首先教学生写一些非常简单的类，接着添加构造函数，随后重载简单操作符，重载 I/O 操作符，依此类推。这种定量教学方式避免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早学会编写现实的类定义，而不是强迫他们花大量时间写一

些故意简化的类。到第 11 章结束时，学生就能写出完整而实用的类，实现“类”的课程目标。

继承问题将在第 10 章进行简单的讲述，使学生能提前接触这一概念。但是，本书要到第 15 章才正式教学生写自己的派生类和使用虚函数。有的授课教师选择在中级课程中再讲述这方面的主题，另一些授课教师则可能想在入门课中就讨论继承问题。如果愿意，有关继承的主题也可推迟讲述，因为第 16 章～第 18 章不需要用到继承。

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，在授课教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”都成为标准的 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。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)。

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(Video Note)

每当看到“Video Note：……”，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://www.aw-bc.com/savitch/videonotes/*。注意，由于视频是英文版的，所以为了方便索引，书中保留了这些视频的英文名称。

课堂实测

世界各地数十万学生都用过本书的前 6 版。许多学生和他们的授课教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本书，表明大多数学生和授课教师都非常喜欢这种写作风格。当然，还有一些读者建议我对本书进行修订。所有修订意见都经过了我仔细的斟酌。正是基于这些宝贵的意见，这一版才能够逐渐成形，最终摆上您的案头。相较于以前各版，这一版能更好地满足学生和授课教师的需要。

主题可以灵活排序

本书允许授课教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。采用推荐的任何一种方式来阅读本书，都不会影响学习的连贯性。当您重新对这些材料进行组织时，为了确保这种连贯性，您可能需要对个别小节的编排顺序进行调整，而保持各章顺序不变。但是，只有处于方便位置的大型小节才需要移动。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，每章都有一个“预备知识”小节，它解释了在继续本章的学习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、函数定义和基本的文件 I/O。第 3 章额外介绍了几种控制结构，如果希望尽早学习类，可以考虑推迟学习这一章的内容。
- **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(也可以推迟到本课程稍晚的时候学习)。
- **指针和链表** 第 13 章。

可能还会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：把类的学习稍微延后一点，但仍然放在前面来进行

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，但比本书的默认顺序还是略微提前一些。

- **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(也可以推迟到本课程稍晚的时候学习)。
- **向量** 8.3 节。
- **指针和链表** 第 13 章。

可能还会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。

支持材料

部分支持材料是本书所有用户都能使用的。另一部分只有符合条件的教师才能使用。

针对本书所有读者

- 本书源代码
- PowerPoint 幻灯片
- 视频注解

为了获取这些材料，请访问 *http://www.aw.com/cssupport*(或者访问本书译者博客 *http://transbot.blog.163.com*)。

针对符合条件的教师

选用本书作为教材的教师，可联系培生(北京)代表处，或者致函 coo@netease.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)。

Preface

This book is meant to be used in a first course in programming and computer science using the C++ language. It assumes no previous programming experience and no mathematics beyond high school algebra.

If you have used the previous edition of this book, you should read the following section that explains the changes to this seventh edition, and then you can skip the rest of this preface. If you are new to this book, the rest of this preface will give you an overview of the book.

Changes to the Seventh Edition

This seventh edition presents the same programming philosophy as the sixth edition. All of the material from the sixth edition remains. Thirty new programming projects have been added, and the presentation has been reworked in several chapters. Additional material on the `map` class has been added and the program examples have been updated for greater compatibility with a majority of compilers in use today. Moreover, thirty-nine videos that cover specific topics and solutions to the Programming Projects have been added to the book's website. These video notes walk students through the process of both problem solving and coding to help reinforce key programming concepts. An icon appears in the margin of the book when a video is available regarding the topic covered in the text. Finally, in response to instructors' reviews of the previous edition, introductory material on inheritance has been moved from Chapter 6, "I/O Streams," to Chapter 10, "Defining Classes." However, if you are an instructor already using the sixth edition, you can continue to teach your course almost without change.

Choose Your Own Ordering of Topics

Most introductory textbooks that use C++ have a very detailed agenda that instructors must follow to use the book in their courses. That is not true of this book. If you are an instructor, this book adapts to the way you teach, rather than making you adapt to the book. You can easily change the order in which chapters and sections are covered without loss of continuity. A dependency chart at the end of this preface gives an overview of the orders in which chapters and sections may be covered, and each chapter has a "Prerequisites" section that explains what parts of the book need to be covered before each section in that chapter. This allows instructors to easily rearrange the order in which chapters, or even sections, are covered.

Although this book uses libraries and teaches students the importance of libraries, it does not require any nonstandard libraries. This book uses only libraries that are provided with essentially all C++ implementations.

Allows either Late or Early Introduction of Classes

This book allows for flexible coverage of classes. You may cover classes either early or late.

The default ordering of chapters begins by introducing students to the fundamental concepts of variable declarations, expression evaluation, control structures, procedural abstraction, function definition, arrays, and pointer manipulation. This gives the student a solid foundation in the most basic of programming concepts before tackling the more sophisticated concepts involved in object-oriented programming. The book covers these most basic of programming concepts in Chapters 2 through 9. In Chapter 6, file I/O streams are used to teach students how to use classes. Because stream I/O does require some use of namespaces and class libraries, some minimal coverage of how to use predefined namespaces and some standard library I/O classes is integrated into the first few chapters. In Chapter 10, students learn how to write their own classes.

Instructors who prefer to introduce classes earlier in the course can rearrange the order of chapters to suit their approach. This is discussed in the section "Flexibility in Topic Ordering" later in this preface. Essentially, instructors can cover Chapter 10, "Defining Classes," immediately after Chapter 6.

The book uses a measured approach to teaching classes. It teaches students to write some very simple classes, then adds constructors, overloading simple operators, overloading the I/O operators, and so forth. This measured approach keeps the student from being overwhelmed with a long list of complicated constructions and concepts. However, one goal of this book is to get students to write realistic class definitions without spending time writing classes that are artificially simple. By the end of Chapter 11, students are writing essentially the same kinds of classes that they will be writing when they finish the course.

Inheritance is covered briefly in Chapter 10 so that students become aware of the concept. However, this book does not teach students how to write their own derived classes and use virtual functions until Chapter 15. Some instructors may choose to leave that material for a second course. Other instructors will want to integrate this inheritance coverage into their course. If desired, the material on inheritance may be moved later, since Chapters 16 through 18 do not require inheritance.

Accessibility to Students

It is not enough for a book to present the right topics in the right order. It is not even enough for it to be clear and correct when read by an instructor or other experienced programmer. The material needs to be presented in a way that is accessible to beginning students. In this introductory textbook, I have endeavored to write in a way that students find clear and friendly. Reports from the many students who have used the earlier editions of this book confirm that this style makes the material clear and often even enjoyable to students.

ANSI/ISO C++ Standard

This edition is fully compatible with compilers that meet the latest ANSI/ISO C++ standard.

Advanced Topics

Many "advanced topics" are becoming part of a standard CS1 course. Even if they are not part of a course, it is good to have them available in the text as enrichment material. This book offers a number of advanced topics that can be integrated into a course or left as enrichment topics. It gives thorough coverage of C++ templates, inheritance (including virtual functions), exception handling, and the STL (Standard Template Library).

Summary Boxes

Each major point is summarized in a boxed section. These boxed sections are spread throughout each chapter.

Self-Test Exercises

Each chapter contains numerous Self-Test Exercises at strategic points. Complete answers for all the Self-Test Exercises are given at the end of each chapter.

Video Notes

The icon in the margin indicates when an online video is available that relates to the current topic in the book.

Classroom Tested

Hundreds of thousands of students have used the first six editions of this book. Many of these students and many of their instructors have given me feedback about what worked and what did not work for them. The vast majority of the comments were extremely positive and indicated that students and teachers liked the book pretty much as it was, but suggestions for some changes were made. All suggestions for changes were carefully considered. That valuable feedback was used to revise this edition so that it fits students' and instructors' needs even better than the previous editions.

Flexibility in Topic Ordering

This book was written to allow instructors wide latitude in reordering the material. To illustrate this flexibility, we suggest two alternative ways to order the topics. There is no loss of continuity when the book is read in either of these ways. To ensure this continuity when you rearrange material, you may need to move sections rather than entire chapters. However, only large sections in convenient locations are moved. To help customize a particular order for any class's needs, the end of this preface contains a dependency chart, and each chapter has a "Prerequisites" section that explains what material needs to be covered before each section in that chapter.

Reordering 1: Earlier Classes

To effectively design classes, a student needs some basic tools such as control structures and function definitions. This basic material is covered in Chapters 1 through 6. After completing Chapter 6, students can begin to write their own classes. One possible reordering of chapters that allows for such early coverage of classes is the following:

Basics: Chapters 1, 2, 3, 4, 5, and 6. This material covers all control structures, function definitions, and basic file I/O. Chapter 3, which covers additional control structures, could be deferred if you wish to cover classes as early as possible.

Classes and namespaces: Chapter 10, Sections 11.1 and 11.2 of Chapter 11, and Chapter 12. This material covers defining classes, friends, overloaded operators, and namespaces.

Arrays, strings and vectors: Chapters 7 and 8.

Pointers and dynamic arrays: Chapter 9.

Arrays in classes: Sections 11.3 and 11.4 of Chapter 11.

Inheritance: Chapter 15.

Recursion: Chapter 14 (Alternately, recursion may be moved to later in the course.)

Pointers and linked lists: Chapter 13.

Any subset of the following chapters may also be used:

Exception handling: Chapter 16.

Templates: Chapter 17.

Standard Template Library: Chapter 18.

Reordering 2: Classes Slightly Later but Still Early

This version covers all control structures and the basic material on arrays before doing classes, but classes are covered later than the previous ordering and slightly earlier than the default ordering.

Basics: Chapters 1, 2, 3, 4, 5, and 6. This material covers all control structures, function definitions, and the basic file I/O.

Arrays and strings: Chapter 7, Sections 8.1 and 8.2 of Chapter 8.

Classes and namespaces: Chapter 10, Sections 11.1 and 11.2 of Chapter 11, and Chapter 12. This material covers defining classes, friends, overloaded operators, and namespaces.

Pointers and dynamic arrays: Chapter 9.

Arrays in classes: Sections 11.3 and 11.4 of Chapter 11.

Inheritance: Chapter 15.

Recursion: Chapter 14. (Alternately, recursion may be moved to later in the course.)

Vectors: Chapter 8.3.

Pointers and linked lists: Chapter 13.

Any subset of the following chapters may also be used:

Exception handling: Chapter 16.

Templates: Chapter 17.

Standard Template Library: Chapter 18.

Support Material

There is support material available to all users of this book and additional material available only to qualified instructors.

Materials Available to All Users of this Book

- Source Code from the book
- PowerPoint slides
- Video Notes

To access these materials, go to:

http://www.aw.com/cssupport

Resources Available to Qualified Instructors Only

Visit Addison-Wesley's instructor resource center (http://www.aw.com/irc), contact your local AW sales representative, or send email to computing@aw.com for information on how to access instructor supplements:

- Instructor's Resource Guide—including chapter-by-chapter teaching hints, quiz questions with solutions, and solutions to many programming projects
- Test Bank and Test Generator
- PowerPoint Lectures—including programs and art from the text
- Lab Manual
- Instructor access to Addison-Wesley's *MyCodeMate*

Visual C++ Support

In addition to the already listed support material, this text is also available with supplementary Visual C++ materials:

- Visual C++ 2008 Programming Companion (includes Microsoft's Visual C++ Express compiler)
- Visual C++ 2005 Programming Companion (includes Microsoft's Visual C++ 2005 compiler)

Contact your local Addison-Wesley/Pearson representative, or send email to computing@aw.com for information on obtaining the book with Visual C++.

Additional Support Material

- Addison-Wesley's *C++ Backpack Reference Guide* is a quick reference to C++'s most frequently used keywords and libraries. It integrates all the specifics of C++ in one handy place including syntax examples, keyword descriptions, and programming tips. The *C++ Backpack Reference Guide* is offered at a discount when packaged with a new copy of this text.
- Addison-Wesley's *MyCodeMate* is a book-specific Web resource that provides tutorial help and evaluation of student programs. **A complimentary subscription to *MyCodeMate* is offered when the access code is ordered in a package with a new copy of this text.** Subscriptions can also be purchased online. For more information, visit http://www.mycodemate.com.

For more information on these and other resources available with the book, contact your local Addison-Wesley representative, or send email to computing@aw.com.

Acknowledgments

Numerous individuals and groups have provided me with suggestions, discussions, and other help in preparing this textbook. Much of the first edition of this book was written while I was visiting the Computer Science Department at the University of Colorado in Boulder. The remainder of the writing on the first edition and the work on subsequent editions was done in the Computer Science and Engineering Department at the University of California, San Diego (UCSD). I am grateful to these institutions for providing a conducive environment for teaching this material and writing this book.

I would like to thank reviewers of this edition. They are: Joseph Allen (Community College of Rhode Island), Noah Aydin (Kenyon College), Andrew Haas (University at Albany, SUNY), Huzefa Kagdi (Kent State University), Gilliean Lee (Lander University), Cynthia Martincic (Saint Vincent College), Jeff Roach (East Tennessee State University), Susanne Sherba (University of Colorado, Boulder), Michal Sramka (Florida Atlantic University), James Stepleton (Stevens Institute of Technology), and Jeffrey Watson (Walla Walla Community College).

I extend a special thanks to all the individuals who have contributed critiques or programming projects for this or earlier editions and drafts of this book. In alphabetical order they are: 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, Dennis Heckman, Bob Holloway, Nisar Hundewale, Matt Johnson, Bruce Johnston, Larry Johnson, Thomas Judson, Paul J. Kaiser, Michael Keenan, Brian R. King, Paul Kube, Barney MacCabe, Steve Mahaney, Michael Main, Walter A. Manrique, Anne Marchant, John Marsaglia, Nat Martin, Bob Matthews, Jesse Morehouse, Ethan Munson, Donald Needham, Dung Nguyen, Joseph D. Oldham, Jennifer Perkins, Carol Roberts, Ken Rockwood, John Russo, Amber Settle, Naomi Shapiro, Scott Strong, David Teague, Jerry Weltman, John J. Westman, and Linda F. Wilson.

I extend a special thanks to the many instructors who used early editions of this book. Their comments provided some of the most helpful reviewing that the book received.

I thank all the individuals at Addison-Wesley who contributed so much to making this book possible: Marilyn Lloyd, Sarah Milmore, Chris Kelly, and my editor, Matt Goldstein.

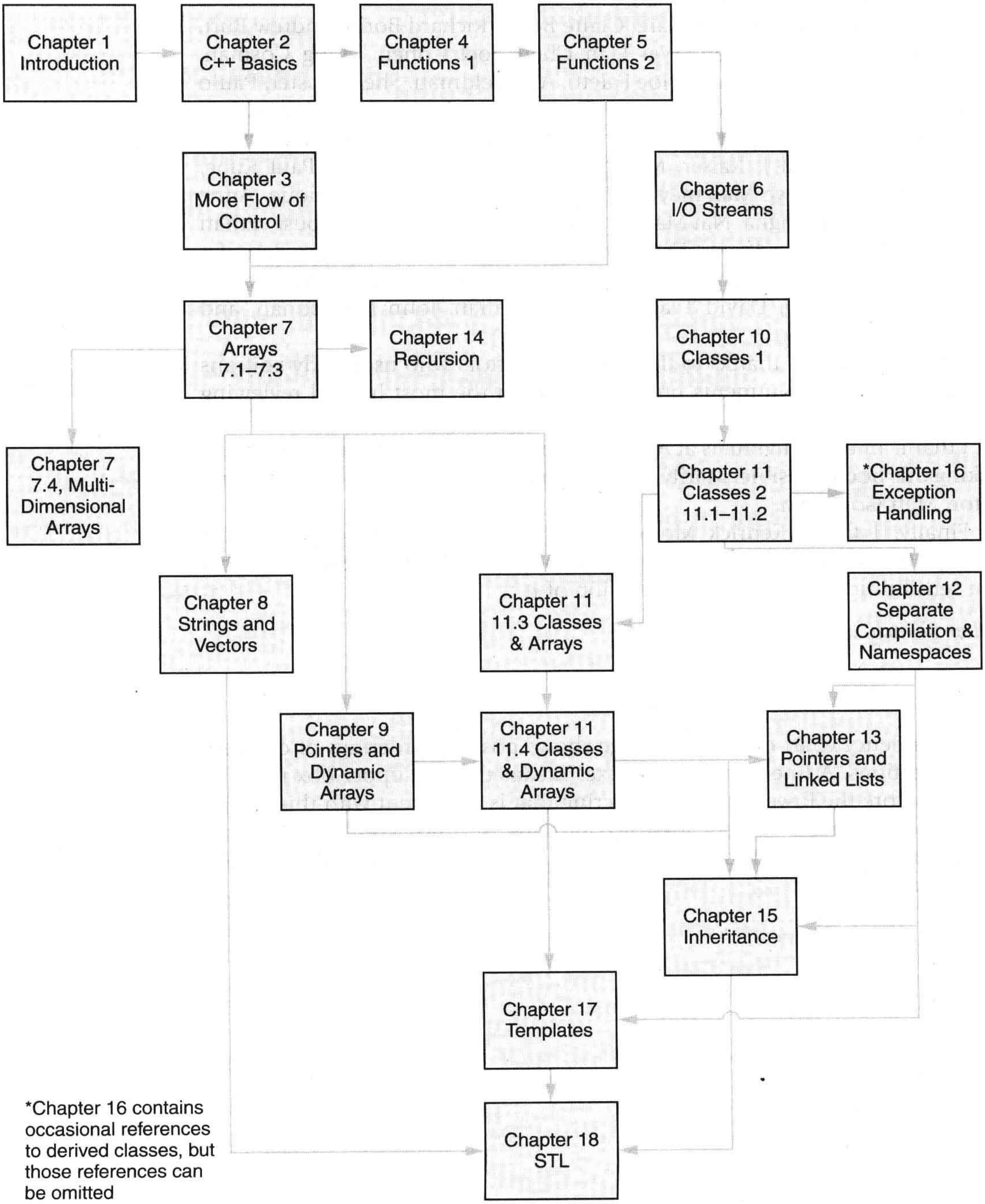
Finally, I thank Kenrick Mock who implemented the changes in this edition. He had the almost impossible task of pleasing me, my editor, and his own sensibilities, and he did a superb job of it.

Walter Savitch

Dependency Chart

The dependency chart on the next page shows possible orderings of chapters and subsections. A line joining two boxes means that the upper box must be covered before the lower box. Any ordering that is consistent with this partial ordering can be read without loss of continuity. If a box contains a section number or numbers, then the box refers only to those sections and not to the entire chapter.

DISPLAY P.1 **Dependency Chart**



*Chapter 16 contains occasional references to derived classes, but those references can be omitted

Brief Contents

Appendices