

PEARSON

- 经久不衰的 C++ 畅销经典教程
- 涵盖 C++11 新标准

C++

Primer Plus (第6版)

英文版 (上册)

[美] Stephen Prata 著

 人民邮电出版社
POSTS & TELECOM PRESS

PEARSON

C++

Primer Plus

(第6版) 英文版 (上册)

[美] Stephen Prata 著

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C++ Primer Plus : 第6版 : 全2册 : 英文 / (美)
普拉达 (Prata, S.) 著. -- 2版. -- 北京 : 人民邮电出版社, 2015.4

ISBN 978-7-115-38111-8

I. ①C… II. ①普… III. ①C语言—程序设计—英文
IV. ①TP312

中国版本图书馆CIP数据核字(2015)第001839号

版权声明

Original edition, C++ Primer Plus (6th Edition), 9780321776402, by Stephen Prata, published by Pearson Education, Inc., publishing as Addison-Wesley Professional, Copyright © 2012.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Inc.

English reprint published by Pearson Education North Asia Limited and Posts & Telecommunication Press, Copyright © 2012.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书封面贴有 Pearson Education 出版集团激光防伪标签, 无标签者不得出售。

-
- ◆ 著 [美] Stephen Prata
 - 责任编辑 傅道坤
 - 责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 89.75
字数: 1 839 千字 2015 年 4 月第 2 版
印数: 3 001 - 6 000 册 2015 年 4 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2012-5224 号
-

定价: 128.00 元 (上、下册)

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

内 容 提 要

C++是在C语言基础上开发的一种集面向对象编程、泛型编程和过程化编程于一体的编程语言，是C语言的超集。本书是根据2003年的ISO/ANSI C++标准编写的，通过大量短小精悍的程序详细而全面地阐述了C++的基本概念和技术，并专辟一章介绍了C++11新增的功能。

全书分18章和10个附录，分别介绍了C++程序的运行方式、基本数据类型、复合数据类型、循环和关系表达式、分支语句和逻辑运算符、函数重载和函数模板、内存模型和名称空间、类的设计和使用、多态、虚函数、动态内存分配、继承、代码重用、友元、异常处理技术、string类和标准模板库、输入/输出、C++11新增功能等内容。

本书针对C++初学者，书中从C语言基础知识开始介绍，然后在此基础上详细阐述C++新增的特性，因此不要求读者有C语言方面的背景知识。本书可作为高等院校C++课程的教材，也可供初学者自学C++时使用。

献给大人们

大人们，你们不要责怪孩子，
他们只是寻找可以玩耍的东西。
你们不要生气，你们可以和他们一起玩，
你们甚至可以和他们一起玩耍。



大人们，你们不要责怪孩子，
他们只是寻找可以玩耍的东西。
你们不要生气，你们可以和他们一起玩，
你们甚至可以和他们一起玩耍。
To my parents, with love.



大人们，你们不要责怪孩子，
他们只是寻找可以玩耍的东西。
你们不要生气，你们可以和他们一起玩，
你们甚至可以和他们一起玩耍。
大人们，你们不要责怪孩子，
他们只是寻找可以玩耍的东西。
你们不要生气，你们可以和他们一起玩，
你们甚至可以和他们一起玩耍。

致谢

本书第 6 版的致谢

对 Pearson 的 Mark Taber 和 Samantha Sinkhorn 在本书编写过程中所做出的指导和管理工作表示感谢，同时还要对 David Horvath 所做的技术审稿和编辑工作表示感谢。

本书第 5 版的致谢

对 Sams Publishing 的 Loretta Yates 和 Songlin Qiu 在本书编写过程中所做出的指导和管理工作表示感谢，同时还要感谢我的同事 Fred Schmitt 为本书提供了实用的建议。我还要再次对 Metrowerks 公司的 Ron Liechty 提供的帮助表示感谢。

本书第 4 版的致谢

Linda Sharp、Karen Wachs 和 Laurie McGuire 这几位来自 Pearson 和 Sams Publishing 的编辑发起了本书的编写工作，并使之坚持下去，向他们表示感谢。感谢 Michael Maddox、Bill Craun、Chris Maunder 和 Phillippe Bruno 所做的技术审稿和编辑工作。还要感谢 Michael Maddox 和 Bill Craun 为 Real Word Notes 提供了资料。最后，还要感谢 Metrowerks 公司的 Ron Liechty 和 Comeau Computing 公司的 Greg Comeau 提供了 C++ 编译器。

本书第 3 版的致谢

谢谢来自 Macmillan 和 The Waite Group 的几位编辑在本书编写过程中所做的贡献。他们是 Tracy Dunkselberger、Susan Walton 和 Andrea Rosenberg。谢谢 Russ Jacobs 所做的文字和技术编辑工作。还要感谢来自 Metrowerks 公司的 Dave Mark、Alex Harper，以及 Ron Liechty，谢谢他们的帮助与合作。

本书第 2 版的致谢

谢谢 Mitchell Waite 和 Scott Calamar 对本书第 2 版的支持，谢谢 Joel Fugazzotto 和 Joanne Miller 从头至尾地指导本书的编写工作。谢谢 Metrowerks 公司的 Michael Marcotty，他们生产的 beat 版本的 CodeWarrior 编译器给我带来了许多困惑，是他帮我一一化解。为本书第 1 版提供反馈意见的下列教师也是我的感谢对象，他们是 Jeff Buckwalter、Earl Brynner、Mike Holland、Andy Yao、Larry Sanders、Shahin Momtazi 和 Don Stephens。最后，我还要感谢 Heidi Brumbaugh，谢谢他为本书新增和修改的内容所做的文字编辑工作。

本书第 1 版的致谢

许多人都对本书贡献良多。我要特别感谢 Mitch Waite，谢谢他对本书所做的扩展、调整和改进工作，以及对本书原稿的审稿工作。Harry Henderson 审读了本书后面几章内容，并使用 Zortech 的 C++ 编译器对书中的程序进行了测试，在此向他表示感谢。感谢 David Gerrold 对整本书所做的审稿工作，并且感谢他支持欠缺经验的读者的需求。Hank Shiffman 使用 Sun C++ 编译器测试了书中的程序，Kent Williams 使用 AT&T cfront 和 G++ 编译器测试了书中的程序，在此向他们二位表示感谢。还要感谢 Borland International 公司的 Nan Borreson 通过 Turbo C++ 和 Boland C++ 提供了及时的帮助。谢谢 Ruth Myers 和 Christine Bush，在本书的编写过程中产生了大量的纸张文件，是你们将其一一处理。最后，感谢 Scott Calamar 让一切工作保持正常运转。

作者简介

Stephen Prata 在美国加州肯特菲尔得的马林学院教授天文、物理和计算机科学。他毕业于加州理工学院，在美国加州大学伯克利分校获得博士学位。他单独或与他人合作编写的编程图书有十多本，其中《New C Primer Plus》获得了计算机出版联合会 1990 年度最佳“*How-to*”计算机图书奖，《C++ Primer Plus》获得了计算机出版联合会 1991 年度最佳“*How-to*”计算机图书奖提名。

Table of Contents

Introduction 1

1 Getting Started with C++ 9

- Learning C++: What Lies Before You 10
- The Origins of C++: A Little History 10
- Portability and Standards 15
- The Mechanics of Creating a Program 18
- Summary 25

2 Setting Out to C++ 27

- C++ Initiation 27
- C++ Statements 41
- More C++ Statements 45
- Functions 48
- Summary 61
- Chapter Review 62
- Programming Exercises 62

3 Dealing with Data 65

- Simple Variables 66
- The `const` Qualifier 90
- Floating-Point Numbers 92
- C++ Arithmetic Operators 97
- Summary 109
- Chapter Review 110
- Programming Exercises 111

4 Compound Types 115

- Introducing Arrays 116
- Strings 120
- Introducing the `string` Class 131
- Introducing Structures 140
- Unions 149
- Enumerations 150
- Pointers and the Free Store 153
- Pointers, Arrays, and Pointer Arithmetic 167
- Combinations of Types 184
- Array Alternatives 186
- Summary 190
- Chapter Review 191
- Programming Exercises 192

- 5 Loops and Relational Expressions 195**
 - Introducing `for` Loops 196
 - The `while` Loop 224
 - The `do while` Loop 231
 - The Range-Based `for` Loop (C++11) 233
 - Loops and Text Input 234
 - Nested Loops and Two-Dimensional Arrays 244
 - Summary 249
 - Chapter Review 250
 - Programming Exercises 251

- 6 Branching Statements and Logical Operators 253**
 - The `if` Statement 254
 - Logical Expressions 260
 - The `cctype` Library of Character Functions 270
 - The `?:` Operator 273
 - The `switch` Statement 274
 - The `break` and `continue` Statements 280
 - Number-Reading Loops 283
 - Simple File Input/Output 287
 - Summary 298
 - Chapter Review 298
 - Programming Exercises 301

- 7 Functions: C++'s Programming Modules 305**
 - Function Review 306
 - Function Arguments and Passing by Value 313
 - Functions and Arrays 320
 - Functions and Two-Dimensional Arrays 337
 - Functions and C-Style Strings 339
 - Functions and Structures 343
 - Functions and `string` Class Objects 353
 - Functions and `array` Objects 355
 - Recursion 357
 - Pointers to Functions 361
 - Summary 371
 - Chapter Review 372
 - Programming Exercises 374

- 8 Adventures in Functions 379**
 - C++ Inline Functions 379
 - Reference Variables 383
 - Default Arguments 409
 - Function Overloading 412
 - Function Templates 419

Introduction

| | |
|-----------------------|-----|
| Summary | 442 |
| Chapter Review | 443 |
| Programming Exercises | 444 |

9 Memory Models and Namespaces 447

| | |
|--------------------------------------|-----|
| Separate Compilation | 447 |
| Storage Duration, Scope, and Linkage | 453 |
| Namespaces | 482 |
| Summary | 497 |
| Chapter Review | 498 |
| Programming Exercises | 501 |

10 Objects and Classes 505

| | |
|---|-----|
| Procedural and Object-Oriented Programming | 506 |
| Abstraction and Classes | 507 |
| Class Constructors and Destructors | 524 |
| Knowing Your Objects: The <code>this</code> Pointer | 539 |
| An Array of Objects | 546 |
| Class Scope | 549 |
| Abstract Data Types | 552 |
| Summary | 557 |
| Chapter Review | 558 |
| Programming Exercises | 559 |

11 Working with Classes 563

| | |
|--|-----|
| Operator Overloading | 564 |
| Time on Our Hands: Developing an Operator | |
| Overloading Example | 565 |
| Introducing Friends | 578 |
| Overloaded Operators: Member Versus Nonmember | |
| Functions | 587 |
| More Overloading: A Vector Class | 588 |
| Automatic Conversions and Type Casts for Classes | 606 |
| Summary | 621 |
| Chapter Review | 623 |
| Programming Exercises | 623 |

12 Classes and Dynamic Memory Allocation 627

| | |
|--|-----|
| Dynamic Memory and Classes | 628 |
| The New, Improved <code>String</code> Class | 647 |
| Things to Remember When Using <code>new</code> | |
| in Constructors | 659 |
| Observations About Returning Objects | 662 |
| Using Pointers to Objects | 665 |
| Reviewing Techniques | 676 |
| A Queue Simulation | 678 |
| Summary | 699 |
| Chapter Review | 700 |
| Programming Exercises | 702 |

Introduction

学习 C++ 是一次探索之旅，因为这种语言容纳了好几种编程范式，其中包括面向对象编程、泛型编程和传统的过程化编程。本书第 5 版是基于 ISO C++ 标准编写的，该标准的官方名称为 C++99 和 C++03 (C++99/C++03)，其中 2003 标准主要是对 1999 标准的技术修正，并没有添加任何新功能。C++ 在不断发展，编写本书时，新标准获得了 C++ 国际标准委员会的批准。在制定期间，该标准名为 C++0x，但现已改名为 C++11。大多数编译器都能很好地支持 C++99/03，而本书的大多数示例也都遵守该标准。有些实现中已显现了新标准的很多特性，而本书也对这些新功能进行了探索。

本书在介绍 C++ 特性的同时，讨论了基本 C 语言，使两者成为有机的整体。书中介绍了 C++ 的基本概念，并通过短小精悍的程序来阐明，这些程序都很容易复制和试验。书中还介绍了输入和输出，如何让程序执行重复性任务，如何让程序做出选择，处理数据的多种方式，以及如何使用函数等内容。另外，本书还讲述了 C++ 在 C 语言的基础上新增的很多特性，包括：

- 类和对象；
- 继承；
- 多态、虚函数和 RTTI（运行阶段类型识别）；
- 函数重载；
- 引用变量；
- 泛型（独立于类型的）编程，这种技术是由模板和标准模板库（STL）提供的；
- 处理错误条件的异常机制；
- 管理函数、类和变量名的名称空间。

初级教程方法

大约 20 年前，《C Primer Plus》开创了优良的初级教程传统，本书建立在这样的基础之上，吸收了其中很多成功的理念。

- 初级教程应当是友好的、便于使用的指南。

- 初级教程不要求您已经熟悉相关的编程概念。
- 初级教程强调的是动手学习，通过简短、容易输入的示例阐述一两个概念。
- 初级教程用示意图来解释概念。
- 初级教程提供问题和练习来检验您对知识的理解，从而适于自学或课堂教学。

基于上述理念，本书帮助您理解这种用途广泛的语言，并学习如何使用它。

- 对何时使用某些特性，例如何时使用公共继承来建立 is-a 关系，提供了概念方面的指导。
- 阐释了常用的 C++ 编程理念和技术。
- 提供了大量的附注，如提示、警告、注意等。

本书的作者和编辑尽最大的努力使本书简单、明了、生动有趣。我们的目标是，您阅读本书后，能够编写出可靠、高效的程序，并且觉得这是一种享受。

示例代码

本书包含大量的示例代码，其中大部分是完整的程序。和前一版一样，本书介绍的是通用 C++，因此适用于任何计算机、操作系统和编译器。书中的示例在 Windows 7 系统、Macintosh OS X 系统和 Linux 系统上进行了测试。使用了 C++11 特性的程序要求编译器支持这些特性，但其他程序可在遵循 C++ 99/03 的任何系统上运行。

书中完整程序的源代码可从配套网站下载，详情请参阅封底的链接信息。

本书内容

本书分为 18 章和 10 个附录。

- **Chapter 1: Getting Started with C++**——本章介绍 Bjarne Stroustrup 如何通过 C 语言的基础上添加对面向对象编程的支持，来创造 C++ 编程语言。讨论面向过程语言（如 C 语言）与面向对象语言（如 C++）之间的区别。您将了解 ANSI/ISO 在制定 C++ 标准方面所做的工作。本章还讨论了创建 C++ 程序的技巧，介绍了当前几种 C++ 编译器使用的方法。最后，本章介绍了本书的一些约定。
- **Chapter 2: Setting Out to C++**——本章介绍创建简单 C++ 程序的步骤。您可以学习到 `main()` 函数扮演的角色以及 C++ 程序使用的一些语句。您将

使用预定义的 `cout` 和 `cin` 对象来实现程序输出和输入，学习如何创建和使用变量。最后，本章还将介绍函数—C++的编程模块。

- **Chapter 3: Dealing with Data**——C++提供了内置类型来存储两种数据：整数（没有小数的数字）和浮点数（带小数的数字）。为满足程序员的各种需求，C++为每一种数据都提供了几个类型。本章将要讨论这些类型，包括创建变量和编写各种类型的常量。另外，还将讨论 C++是如何处理不同类型之间的隐式和显式转换的。
- **Chapter 4: Compound Type**——C++让程序员能够使用基本的内置类型来创建更复杂的类型。最高级的形式是类，这将在第 9 章~第 13 章讨论。本章讨论其他形式，包括数组（存储多个同类型的值）、结构（存储多个不同类型的值）、指针（标识内存位置）。您还将学习如何创建和存储文本字符串及如何使用 C 风格字符数组和 C++的 `string` 类来处理文本输入和输出。最后，还将学习 C++处理内存分配的一些方法，其中包括用于显式地管理内存的 `new` 和 `delete` 运算符。
- **Chapter 5: Loops and Relational Expressions**——程序经常需要执行重复性操作，为此 C++提供了 3 种循环结构：`for` 循环、`while` 循环和 `do while` 循环。这些循环必须知道何时终止，C++的关系运算符使程序员能够创建测试来引导循环。本章还将介绍如何创建逐字符地读取和处理输入的循环。最后，您将学习如何创建二维数组以及如何使用嵌套循环来处理它们。
- **Chapter 6: Branching Statements and Logical Operators**——如果程序可以根据实际情况调整执行，我们就说程序能够智能地行动。在本章，您将了解到如何使用 `if`、`if else` 和 `switch` 语句及条件运算符来控制程序流程，学习如何使用逻辑运算符来表达决策测试。另外，本章还将介绍确定字符关系（如测试字符是数字还是非打印字符）的函数库 `cctype`。最后，还将简要地介绍文件输入/输出。
- **Chapter 7: Functions: C++'s Programming Modules**——函数是 C++的基本编程部件。本章重点介绍 C++函数与 C 函数共同的特性。具体地说，您将复习函数定义的通用格式，了解函数原型是如何提高程序可靠性的。同时，还将学习如何编写函数来处理数组、字符串和结构。还要学习有关递归的知识（即函数在什么情况下调用自身）以及如何用它来实现分而治

之的策略。最后将介绍函数指针，它使程序员能够通过函数参数来命令函数使用另一个函数。

- **Chapter 8: Adventures in Functions**——本章将探索 C++ 中函数新增的特性。您将学习内联函数，它可以提高程序的执行速度，但会增加程序的长度；还将使用引用变量，它们提供了另一种将信息传递给函数的方式。默认参数使函数能够自动为函数调用中省略的函数参数提供值。函数重载使程序员能够创建多个参数列表不同的同名函数。类设计中经常使用这些特性。另外，您还将学习函数模板，它们使程序员能够指定相关函数族的设计。

- **Chapter 9: Memory Models and Namespaces**——第本章讨论如何创建多文件程序，介绍分配内存的各种方式、管理内存的各种方式以及作用域、链接、名称空间，这些内容决定了变量在程序的哪些部分是可见的。

- **Chapter 10: Objects and Classes**——类是用户定义的类型，对象（如变量）是类的实例。本章介绍面向对象编程和类设计。对象声明描述的是存储在对象中的信息以及可对对象执行的操作（类方法）。对象的某些组成部分对于外界来说是可见的（公有部分），而某些部分却是隐藏的（私有部分）。特殊的类方法（构造函数和析构函数）在对象创建和释放时发挥作用。在本章中，您将学习所有这些内容以及其他类知识，了解如何使用类来实现 ADT，如栈。

- **Chapter 11: Working with Classes**——在本章中，您将深入了解类。首先了解运算符重载，它使程序员能够定义与类对象一起使用的运算符，如+。还将学习友元函数，这些函数可以访问外部世界不可访问的类数据。同时还了解一些构造函数和重载运算符成员函数是如何被用来管理类类型转换的。

- **Chapter 12: Classes and Dynamic Memory Allocation**——一般来说，让类成员指向动态分配的内存很有用。如果程序员在类构造函数中使用 new 来分配动态内存，就有责任提供适当的析构函数，定义显式拷贝构造函数和显式赋值运算符。本章介绍了在程序员没有提供显式定义时，将如何隐式地生成成员函数以及这些成员函数的行为。您还将通过使用对象指针，了解队列模拟问题，扩充类方面的知识。

- **Chapter 13: Class Inheritance**——在面向对象编程中，继承是功能最强大

的特性之一，通过继承，派生类可以继承基类的特性，可重用基类代码。本章讨论公有继承，这种继承模拟了 is-a 关系，即派生对象是基对象的特例。例如，物理学家是科学家的特例。有些继承关系是多态的，这意味着相同的方法名称可能导致依赖于对象类型的行为。要实现这种行为，需要使用一种新的成员函数—虚函数。有时，使用抽象基类是实现继承关系的最佳方式。本章讨论了这些问题，说明了公有继承在什么情况下合适，在什么情况下不合适。

- **Chapter 14: Reusing Code in C++**——公有继承只是代码重用的方式之一。本章将介绍其他几种方式。如果一个类包含了另一个类的对象，则称为包含。包含可以用来模拟 has-a 关系，其中一个类包含另一个类的对象。例如，汽车有马达。也可以使用私有继承和保护继承来模拟这种关系。本章说明了各种方法之间的区别。同时，您还将学习类模板，它让程序员能够使用泛型定义类，然后使用模板根据具体类型创建特定的类。例如，栈模板使程序员能够创建整数栈或字符串栈。最后，本章还将介绍多重公有继承，使用这种方式，一个类可以从多个类派生而来。
- **Chapter 15: Friends, Exceptions, and More**——本章扩展了对友元的讨论，讨论了友元类和友元成员函数。然后从异常开始介绍了 C++ 的几项新特性。异常为处理程序异常提供了一种机制，如函数参数值不正确或内存耗尽等。您还将学习 RTTI，这种机制用来确定对象类型。最后，本章还将介绍一种更安全的方法来替代不受限制的强制类型转换。
- **Chapter 16: The string Class and the Standard Template Library**——本章讨论 C++ 语言中新增的一些类库。对于传统的 C-风格字符串来说，string 类是一种方便且功能强大的替代方式。auto_ptr 类帮助管理动态分配的内存。STL 提供了几种类容器（包括数组、队列、链表、集合和映射）的模板表示。它还提供了高效的泛型算法库，这些算法可用于 STL 容器，也可用于常规数组。模板类 valarray 为数值数组提供了支持。
- **Chapter 17: Input, Output, and Files**——本章复习 C++ 的 I/O，并讨论如何格式化输出。您将要学习如何使用类方法来确定输入或输出流的状态，

了解输入类型是否匹配或是否检测到了文件尾。C++ 使用继承来派生用于管理文件输入和输出的类。您将学习如何打开文件，以进行输入和输出，如何在文件中追加数据，如何使用二进制文件，如何获得对文件的随机访问权。最后，还将学习如何使用标准的 I/O 方法来读取和写入字符串。

- **Chapter 18: Visiting with the New C++ Standard**——本章首先复习之前介绍过的几项 C++11 新功能，包括新类型、统一的初始化语法、自动类型推断、新的智能指针以及作用域内枚举。然后，讨论新增的右值引用类型以及如何使用它来实现移动语义。接下来，介绍了新增的类功能、lambda 表达式和可变参数模板。最后，概述了众多其他的新功能。
- **Appendix A: Number Bases**——本附录讨论八进制数、十六进制数和二进制数。
- **Appendix B: C++ Reserved Words**——本附录列出了 C++ 关键字。
- **Appendix C: The ASCII Character Set**——本附录列出了 ASCII 字符集及其十进制、八进制、十六进制和二进制表示。
- **Appendix D: Operator Precedence**——本附录按优先级从高到低的顺序列出了 C++ 的运算符。
- **Appendix E: Other Operators**——本附录总结了正文中没有介绍的其他 C++ 运算符，如按位运算符等。
- **Appendix F: The string Template Class**——本附录总结了 string 类方法和函数。
- **Appendix G: The Standard Template Library Methods and Functions**——本附录总结了 STL 容器方法和通用的 STL 算法函数。
- **Appendix H: Selected Reading and Internet Resources**——本附录列出一些参考书，帮助您深入了解 C++。
- **Appendix I: Converting to ISO Standard C++**——本附录提供了从 C 和老式 C++ 实现到标准 C++ 的转换指南。
- **Appendix J: Answers to Chapter Review**——本附录提供各章结尾的复习题的答案。

对教师的提示

本书宗旨之一是，提供一本既可用于自学又可用于教学的书籍。下面是本书在支持教学方面的一些特征。

- 本书介绍的是通用 C++，不依赖于特定实现。
- 本书内容跟踪了 ISO/ANSI C++ 标准委员会的工作，并讨论了模板、STL、string 类、异常、RTTI 和名称空间。
- 本书不要求学生了解 C 语言，但如果有一定的编程经验则更好。
- 本书内容经过了精心安排，前几章可以作为对 C 预备知识的复习一带而过。
- 各章都有复习题和编程练习。附录 J 提供了复习题的答案。
- 本书介绍的一些主题很适于计算机科学课程，包括抽象数据类型 (ADT)、栈、队列、简单链表、模拟、泛型编程以及使用递归来实现分而治之的策略。
- 各章都非常简短，用一周甚至更短的时间就可以学完。
- 本书讨论了何时使用具体的特性以及如何使用它们。例如，把 is-a 关系的公有继承同组合、has-a 关系的私有继承联系起来，讨论了何时应使用虚函数以及何时不应使用。

本书约定

为区别不同类型的文本，我们使用了一些印刷上的约定。

- 代码行、命令、语句、变量、文件名和程序输出使用 **courier new** 字体：

```
#include <iostream>
int main()
{
    using namespace std;
    cout << "What's up, Doc!\n";
    return 0;
}
```

- 用户需要输入的程序输入用粗体表示：

Please enter your name:

Plato

- 语法描述中的占位符用斜体表示。您应使用实际的文件名、参数等替换占位符。
- 新术语用斜体表示。