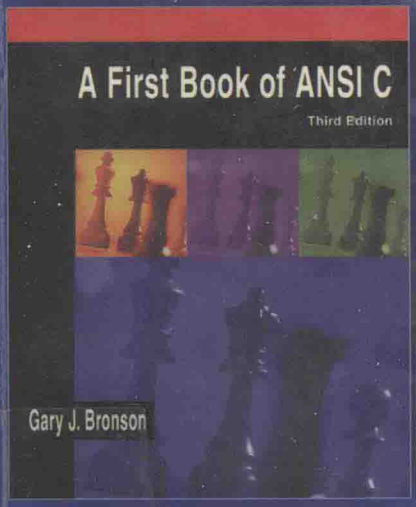
国外计算机科学教材系列

标准C语言基础教程

（第三版）

A First Book of ANSI C

Third Edition



英文版

[美] Gary J. Bronson 著

THOMSON

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

国外计算机科学教材系列

标准C语言基础教程

（第三版）

（英文版）

A First Book of ANSI C

Third Edition

［美］Gary J. Bronson 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

这是一本介绍C语言编程的经典教科书。作者从不同的角度出发，系统介绍了C语言编程中涉及到的数据类型、算术运算、逻辑运算、变量、条件语句、函数、数组、指针、字符串、结构、文件操作、位操作、宏、库函数等基本内容，使读者在学习之后就能很快地掌握C语言编程的精髓。本书还专门介绍了C++的一些重要特性，如抽象数据类型、类和类的继承等。书中每章都附有大量习题，附录还列出了主要习题的答案。

本书条块清晰，章节设置合理，适合用做高等学校本科或专科教材，也适合于初学编程的自学者。

981-254-689-8

English reprint Copyright © 2004 by Thomson Learning and Publishing House of Electronics Industry.
A First Book of ANSI C, Third Edition by Gray J. Bronson, Copyright © 2001. First published by Brooks/Cole, a division of Thomson Learning, Inc(www.thomsonlearningasia.com).
All Rights Reserved.
Authorized Reprint Edition by Thomson Learning and Publishing House of Electronics Industry. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and Publishing House of Electronics Industry.

本书英文影印版由电子工业出版社和汤姆森学习出版集团合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号　图字：01-2004-4587

图书在版编目（CIP）数据

标准C语言基础教程 = A First Book of ANSI C：第三版 /（美）布朗森（Bronson, G. J.）著.
-北京：电子工业出版社，2004.9
（国外计算机科学教材系列）
ISBN 7-121-00250-7

Ⅰ.标... Ⅱ.布... Ⅲ.C语言-程序设计-教材-英文 Ⅳ. TP312

中国版本图书馆CIP数据核字（2004）第083549号

责任编辑：谭海平　许菊芳
印　　刷：北京兴华印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：35.25　字数：790千字
印　　次：2004年9月第1次印刷
定　　价：45.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

主　任　杨芙清　北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长

委　员　王　珊　中国人民大学信息学院院长、教授

胡道元　清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表

钟玉琢　清华大学计算机科学与技术系教授
中国计算机学会多媒体专业委员会主任

谢希仁　中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师

尤晋元　上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任

施伯乐　上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长

邹　鹏　国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员

张昆藏　青岛大学信息工程学院教授

试读结束：需要全本请在线购买：　www.ertongbook.com

Preface

A First Book of ANSI C is written for students who need to know the ANSI C language and how to write its syntax. The pedagogy helps students with common programming pitfalls, which makes this book more useful to a student than the typical reference-type book. It assumes no programming background, but can be used by someone who wants to learn this language. The primary purpose of the first two editions of this text was to make C more accessible as an applications programming language than was possible with texts that focused on C as an advanced operating systems language. The success of these editions and the many comments received from students and faculty stating that the book really did help them learn and teach C, respectively, have been extremely gratifying.

Based on suggestions and more in-depth responses from adopters, numerous new pedagogical features and material have been incorporated into this third edition. The most noticeable of these changes is that, in all program examples, `main`'s header line has been changed from `void main(void)` to `int main()`, and that the `main` function always returns a value to the operating system. This change was made to reflect current programming practice that the `main` function explicitly return a value. Additional changes to this edition include the following:

- New material on function templates has been added.
- Programming Notes, a single highlighted and informational reference, incorporates the previous Closer Look and Tips From the Pros boxes as well as new material.
- Information on creating C programs under both the Microsoft® Visual C++ 6.0 and the Borland® C++ Builder replaces previous information on outdated C compilers.
- All function prototypes are global.
- A chapter supplement on random number generation has been added.
- New material on C++'s Standard Template Library has been included.

To facilitate using C as a basis for learning C++, the two chapters introducing C++ that were introduced in the second edition have been retained and updated to ANSI C++ specifications. Thus, as with the second edition, this text can be used as an introduction to programming in general, as an introduction to the C language in particular, and as a basis for further study of the C++ language.

The basic requirement of this third edition, however, remains the same as the first two editions: that is, that all topics be presented in a clear, unambiguous, and accessible manner to beginning students. Toward this end, the heart of the first two editions, which consisted of Chapters 1 through 11, remains essentially unchanged in the present edition. Thus, all of the topics, examples, explanations, and figures in the first 11 chapters of the previous editions remain in the third edition with the addition of the new material.

Distinctive Features of This Book

Writing Style. I firmly believe that introductory texts do not teach students—professors teach students. An introductory textbook, if it is to be useful, must be the primary "supporting actor" to the "leading role" of the professor. Once the professor sets the stage, however, the textbook must encourage, nurture, and assist the student in acquiring and "owning" the material presented in class. To do this the text must be written in a manner that makes sense to the student. My primary

concern, and one of the distinctive features of this book, is that it has been written for the student. As one of my reviewers has said of the first edition, "This book addresses the student and not the professional." Thus, first and foremost, I feel the writing style used to convey the concepts presented is the most important aspect of the text.

Modularity. C, by its nature, is a modular language. Thus, the connection between C functions and modules is made early in the text, in Section 1.2, and continues throughout the book. To stress the modular nature of C, the first complete `main()` function illustrates calling four other functions. The first program that can be compiled, which calls the `printf()` function, is then presented.

The idea of argument passing into modules is also made early, in Section 1.3, with the use of the `printf()` function. In this manner, students are introduced to functions and argument passing as a natural technique of programming.

Software Engineering. As with the first two editions, this revised edition introduces students to the fundamentals of software engineering right from the start. This introduction begins in Section 1.1, which introduces algorithms and the various ways that an algorithm can be described. The example illustrating three algorithms for summing the numbers from 1 to 100 (Figure 1.4) has been retained from the earlier editions.

The increased emphasis on software engineering is supported in the text, starting with Section 1.5, which introduces top-down program development. Here the importance of understanding the problem and selecting an appropriate algorithm is highlighted and the relationship between analysis, design, coding, and testing is introduced. Problem solving within this context is stressed throughout the text.

Introduction to Pointers. One of the unique features of the first edition was the early introduction of pointer concepts. This was done by simply using the `printf()` function initially to display the addresses of variables, and then using variables to store addresses. This approach always seemed a more logical and intuitive method of understanding pointer variables than the indirection description in vogue at the time the first edition was released.

Since the first edition I have been pleased to see that the use of the `printf()` function to display addresses has become a standard way of introducing pointers. Although this approach, therefore, is no longer a unique feature of my book, I am very proud of its presentation and continue to use it in this new edition.

Program Testing. Every C program in this text has been successfully compiled and run under the Microsoft Visual C++ 6.0 compiler. All programs have been written following the current ANSI C standard. Source code for all program examples used in the text is available online. This permits students both to experiment with and extend the existing programs and to modify them more easily as required by a number of end-of-section exercises.

Pedagogical Features

To facilitate my goal of making C accessible as a first-level course, the text includes the following pedagogical features.

End of Section Exercises. Almost every section in the book contains numerous and diverse skill builder and programming exercises. Additionally, solutions to selected odd-numbered exercises are provided in an appendix.

Pseudocode and Flowchart Descriptions. As in the first two editions, pseudocode is stressed throughout the text. Material on flowchart symbols and the use of flowcharts in visually presenting flow-of-control constructs is also presented.

Common Programming Errors and Chapter Review. Each chapter ends with a section on common programming errors and a review of the main topics covered in the chapter.

Programming Notes. A set of shaded boxes that highlight important concepts and useful technical points and programming techniques used by professional programmers is provided.

Enrichment Sections. Given the many different emphases that can be applied in teaching C, a number of Enrichment Sections have been included as chapter supplements. These allow you to provide different emphasis with different students or different C class sections.

Appendices and Supplements. As with the first two editions, an expanded set of appendices is provided in this third edition. These include appendices on operator precedence; ASCII codes; I/O and standard error redirection; the Standard C Library; the Standard Template Library (STL); program entry, compilation, and execution; using the Microsoft Visual C++ 6.0 compiler; and using the Borland C++ Builder compiler.

A final appendix offers solutions to selected odd-numbered problems. Source code for all program examples used in the text is available at

http://www.brookscole.com

An instructor's manual including chapter outlines and answers to selected even-numbered problems is also available.

Acknowledgments

The writing of this third edition is a direct result of the success (and limitations) of the first two editions. In this regard, my most heartfelt acknowledgment and appreciation is to the instructors and students who found these editions to be of service to them in their respective quests to teach and learn C.

Once a third edition was planned, its completion depended on the encouragement, skills, and efforts of many other people. For this I especially want to thank the staff of Brooks/Cole Publishing Company for their many contributions. First and foremost, this includes my project editor, Kallie Swanson. Additionally, I am very grateful to editorial assistants Grace Fujimoto and Meg Weist for handling numerous scheduling and review details that permitted me to concentrate on the actual writing of the text.

I also wish to express my gratitude to the individual reviewers:

Taret Alameldin, California State University at Fresno
John Avitabile, College of Saint Rose
Larry Booth, Highline Community College
Li-hsiang Cheo, William Paterson University
Yonina S. Cooper, University of Las Vegas
Ralph Ewton, University of Texas at El Paso
James Gips, Boston College
Richard A. Hatch, San Diego State University
Rick L. Homkes, Indiana University at Kokomo
Michael Milligan, Front Range Community College
Robert E. Norton, San Diego Mesa College
Robert Probasco, University of Idaho
Ashraf Saad, University of Cincinnati
Anthony Sako, Columbia Basin College
Robert Weaver, Mt. Holyoke College
Lee Westbrock, Trident Technical College
Bob Willis, Rogers State University

Each of them supplied extremely detailed and constructive reviews of both the original manuscript and a number of revisions. Their suggestions, attention to detail, and comments were extraordinarily helpful in the preparation of the various editions.

Finally, the task of turning the final manuscript into a textbook again required a dedicated production staff. For this I especially want to thank copy editor Philip Jochnowitz, production editor Robin Gold of Forbes Mill Press, manuscript keyer Sue Boshers, and compositor Linda Weidemann of Wolf Creek Press. Their dedication, attention to detail, and their high standards have helped immensely to improve the quality of this edition. Almost from the moment the book moved to the production stage they seemed to take personal ownership of the text, and I am very grateful to them.

Special thanks also go to Janie Schwark, Academic Product Manager of Developer Tools at Microsoft, for providing invaluable support and product information.

I also gratefully acknowledge the direct encouragement and support provided by my dean Dr. Paul Lerman, my associate dean, Dr. Ron Heim, and my chairperson, Dr. Youngboem Kim. Without their support, this text could not have been written.

Finally, I deeply appreciate the patience, understanding, and love provided by my wife, friend, and partner, Rochelle.

Gary Bronson

Contents

Part 1 Fundamentals 1

Chapter 1 Getting started 1

1.1 Introduction to Programming 1

Algorithms 3

From Algorithms to Programs 6

Program Translation 6

Exercises 1.1 7

1.2 Introduction to Modularity 8

Functions 9

The `main()` Function 10

Exercises 1.2 12

1.3 The `printf()` Function 13

Exercises 1.3 16

1.4 Programming Style 16

Comments 17

Exercises 1.4 19

1.5 Top-Down Program Development 20

Modularity and Top-Down Design 22

Exercises 1.5 23

1.6 Common Programming Errors 25

1.7 Chapter Summary 26

Chapter 2 Data Types, Declarations, and Displays 27

2.1 Data Types 27

Integer Values 27

Floating Point and Double Precision Numbers 28

Exponential Notation 28

Character Type 28

Escape Sequences 29

Exercises 2.1 30

2.2 Arithmetic Operators 31

Integer Division 31

A Unary Operator (Negation) 32

Operator Precedence and Associativity 32

Exercises 2.2 33

2.3 Displaying Numerical Results 34
Formatted Output 36
Format Modifiers 38
Other Number Bases 38
Exercises 2.3 41
2.4 Variables and Declarations 43
Declaration Statements 45
Declaration Statements as Definition Statements 48
Exercises 2.4 49
2.5 Integer Qualifiers 52
Determining Storage Size 53
Exercises 2.5 54
2.6 Common Programming Errors 54
2.7 Chapter Summary 55
2.8 Chapter Supplement: Bits, Bytes, Addresses, and Number Codes 56
Words and Addresses 56
Two's Complement Numbers 56

Chapter 3 Assignments, Addresses, and Interactive Input 58

3.1 Assignment 58
Assignment Operators 61
Accumulating 62
Counting 63
Exercises 3.1 65
3.2 Addresses 66
Storing Addresses 68
Using Addresses 69
Declaring Pointers 69
Exercises 3.2 71
3.3 The `scanf()` Function 74
Exercises 3.3 76
3.4 `scanf()` with Buffered Input 80
3.5 Symbolic Constants 82
Exercises 3.5 85
3.6 Common Programming Errors 85
3.7 Chapter Summary 86
3.8 Chapter Supplement: Introduction to Abstraction 87
Built-in and Abstract Data Types 88
Procedural Abstraction 88

Part 2 Flow of Control 89

Chapter 4 Selection 89

4.1 Relational Expressions 89
Logical Operators 91

试读结束：需要全本请在线购买： www.ertongbook.com

A Numerical Accuracy Problem 92
Exercises 4.1 93
4.2 The `if-else` Statement 93
Compound Statements 96
One-Way Selection 97
Problems Associated with the `if-else` Statement 98
Exercises 4.2 99
4.3 Nested if Statements 101
The `if-else` Chain 102
Exercises 4.3 104
4.4 The `switch` Statement 106
Exercises 4.4 109
4.5 Common Programming Errors 110
4.6 Chapter Summary 111
4.7 Chapter Supplement: Errors, Testing, and Debugging 113
Compile-Time and Run-Time Errors 113
Syntax and Logic Errors 114
Testing and Debugging 115

Chapter 5 Repetition 118
5.1 The `while` Statement 118
Exercises 5.1 123
5.2 `scanf()` within a `while` Loop 124
Sentinels 129
`break` and `continue` Statements 131
The Null Statement 132
Exercises 5.2 132
5.3 The `for` Statement 134
`scanf()` within a `for` Loop 138
Nested Loops 139
Exercises 5.3 141
5.4 The `do` Statement 144
Validity Checks 145
Exercises 5.4 146
5.5 Common Programming Errors 146
5.6 Chapter Summary 147
5.7 Chapter Supplement: Character-Oriented Graphics 148
Character-at-a-time Display 148
Exercises 5.7 151

Part 3 Functions 152

Chapter 6 Writing Your Own Functions 152
6.1 Function Definitions and Declarations 152

Defining a Function 153
Function Prototypes 156
Calling a Function 157
Placement of Statements 160
Function Stubs 161
Functions with Empty Parameter Lists 163
Exercises 6.1 163
6.2 Standard Library Functions 166
Input/Output Library Functions 167
Mathematical Library Functions 167
Casts 169
String Library Functions 170
Miscellaneous Routines 170
Exercises 6.2 171
6.3 Variable Scope 172
Misuse of Global Variables 174
Exercises 6.3 175
6.4 Variable Storage Class 177
Local Variable Storage Classes 177
Global Variable Storage Classes 180
Exercises 6.4 182
6.5 Passing Addresses 183
Passing, Storing, and Using Addresses 184
Exercises 6.5 188
6.6 Common Programming Errors 190
6.7 Chapter Summary 190
6.8 Chapter Supplement: Generating Random Numbers 191
Scaling 193

Part 4 Complex Data Types 194

Chapter 7 Arrays 194
7.1 Single-Dimensional Arrays 195
Input and Output of Array Values 198
Exercises 7.1 201
7.2 Array Initialization 202
Exercises 7.2 204
7.3 Passing Arrays 205
Exercises 7.3 209
7.4 Two-Dimensional Arrays 209
Larger Dimensional Arrays 215
Exercises 7.4 216
7.5 Common Programming Errors 217
7.6 Chapter Summary 217

7.7 Chapter Supplement: Searching and Sorting Methods 218
Search Algorithms 218
Linear Search 218
Return the index value 219
Binary Search 220
Big O Notation 224
Sort Algorithms 224
The Selection Sort 224
An Exchange ("Bubble") Sort 227

Chapter 8 Arrays, Addresses, and Pointers 230
8.1 Array Names as Pointers 230
Exercises 8.1 234
8.2 Pointer Arithmetic 235
Pointer Initialization 238
Exercises 8.2 238
8.3 Passing and Using Array Addresses 239
Advanced Pointer Notation 242
Exercises 8.3 243
8.4 Common Programming Errors 245
8.5 Chapter Summary 246

Chapter 9 Character Strings 248
9.1 String Fundamentals 248
String Input and Output 248
String Processing 250
Character-by-Character Input 251
Exercises 9.1 253
9.2 Pointers and Library Functions 254
Library Functions 258
Exercises 9.2 260
9.3 String Definitions and Pointer Arrays 261
Pointer Arrays 263
Exercises 9.3 266
9.4 Formatting Strings 267
In-Memory String Conversions 268
Format Strings 269
Exercises 9.4 269
9.5 Common Programming Errors 270
9.6 Chapter Summary 270

Chapter 10 Structures 271
10.1 Single Structures 272
Exercises 10.1 275

10.2 Arrays of Structures 276
Exercises 10.2 279
10.3 Passing and Returning Structures 280
Returning Structures 283
Exercises 10.3 284
10.4 Linked Lists 285
Exercises 10.4 291
10.5 Dynamic Memory Allocation 292
Dynamically Linked Lists 296
Exercises 10.5 301
10.6 Unions 302
Exercises 10.6 303
10.7 Common Programming Errors 304
10.8 Chapter Summary 304

Part 5 Additional Topics 305

Chapter 11 Data Files 305
11.1 Declaring, Opening, and Closing Files 305
Opening a File 306
Closing a File 309
Exercises 11.1 309
11.2 Reading and Writing Files 310
Standard Device Files 314
Other Devices 315
Exercises 11.2 315
11.3 Random File Access 317
Exercises 11.3 319
11.4 Passing and Returning File Names 320
Exercises 11.4 322
11.5 Common Programming Errors 322
11.6 Chapter Summary 323
11.7 Chapter Supplement: Control Codes 323

Chapter 12 Additional Capabilities 326
12.1 Additional Features 326
The typedef Declaration Statement 326
The `enum` Specifier 327
Casts 330
Conditional Expressions 331
The `goto` Statement 332
12.2 Bitwise Operations 333
The `AND` Operator 333
The Inclusive `OR` Operator 334
The Exclusive `OR` Operator 335

The Complement Operator 336
Different-Sized Data Items 336
The Shift Operators 337
12.3 Macros 339
12.4 Command-Line Arguments 341
A showad three blind mice 341
Exercises 344
12.5 Chapter Summary 346

Part 6 Introduction to C++ 348

Chapter 13 Introduction to C++ 348
13.1 Object-Based Programming and Abstract Data Types 348
Abstract Data Types 349
Comments in C++ 354
Exercises 13.1 354
13.2 Input and Output in C++ 355
Output in C++ 355
Formatted Output 3 357
Input in C++ 361
Exercises 13.2 362
13.3 Function Enhancements in C++ 364
Default Arguments 364
Reusing Function Names (Overloading) 365
Function Templates6 366
Inline Functions7 369
Reference Parameters 371
Passing and Using References 373
References versus Pointers 375
Exercises 13.3 375
13.4 Abstract Data Types in C++ (Classes) 376
Class Construction 376
Terminology 383
Exercises 13.4 383
13.5 Constructors and Destructors 385
Calling Constructors 387
Overloaded and Inline Constructors 388
Destructors 391
Arrays of Objects 391
Exercises 13.5 393
13.6 An Application 394
Exercises 13.6 397
13.7 Common Programming Errors 397
13.8 Chapter Summary 398

Chapter 14 Additional Class Capabilities 400
14.1 Assignment 400
Copy Constructors 403
Base/Member Initialization 405
Exercises 14.1 406
14.2 Additional Class Features 407
Class Scope 407
Static Class Members 408
The `this` Pointer 411
Friend Functions 413
Exercises 14.2 414
14.3 Operator Functions 414
The Assignment Operator Revisited 420
Exercises 14.3 422
14.4 Data Type Conversions 423
Built-in to Built-in Conversion 424
Built-in to User-defined (Class) Conversion 424
User-defined (Class) to Built-in Conversion 426
User-defined (Class) to User-defined (Class) Conversion 427
Exercises 14.4 430
14.5 Class Inheritance 430
Inheritance 431
Access Specifications 432
An Example 433
Polymorphism 436
Exercises 14.5 439
14.6 Common Programming Errors 440
14.7 Chapter Summary 441

Appendix A Operator Precedence Table 443

Appendix B ASCII Character Codes 445

Appendix C Input, Output, and Standard Error Redirection 446

Appendix D The Standard C Library 448

Appendix E The Standard Template Library (C++) 452

Appendix F Program Entry, Compilation, and Execution 457

Appendix G Using Visual C++ Version 6.0 462

Appendix H Using C++ Builder 470

Appendix I Solutions 472

试读结束：需要全本请在线购买： www.ertongbook.com