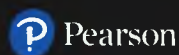


国外计算机科学教材系列



**C How to Program**

International Edition, Seventh Edition

# C 语言大学教程

(第七版)(英文版)

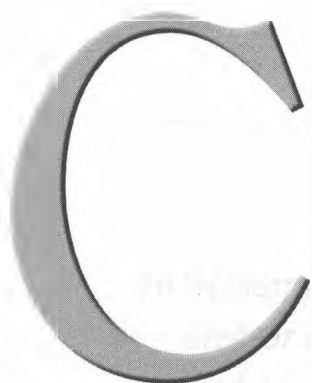
[美] Paul Deitel  
Harvey Deitel 著



中国工信出版集团



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>



# HOW TO PROGRAM

SEVENTH EDITION

**Paul Deitel**

*Deitel & Associates, Inc.*

**Harvey Deitel**

*Deitel & Associates, Inc.*

*International Edition contributions by*

*Mobit P. Tabiliani*

*National Institute of Technology Karnataka,  
Surathkal*

**PEARSON**

Boston Columbus Indianapolis New York San Francisco Upper Saddle River  
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto  
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: *Marcia J. Horton*  
Editor-in-Chief: *Michael Hirsch*  
Associate Editor: *Carole Snyder*  
Vice President, Marketing: *Patrice Jones*  
Marketing Manager: *Yezan Alayan*  
Marketing Coordinator: *Kathryn Ferranti*  
Vice President, Production: *Vince O'Brien*  
Managing Editor: *Jeff Holcomb*  
Associate Managing Editor: *Robert Engelhardt*  
Publisher, International Edition: *Angshuman Chakraborty*  
Acquisitions Editor, International Edition: *Somnath Basu*  
Publishing Assistant, International Edition: *Shokhi Shah*  
Print and Media Editor, International Edition: *Ashwitha Jayakumar*  
Project Editor, International Edition: *Jayashree Arunachalam*  
Publishing Administrator, International Edition: *Hema Mehta*  
Senior Manufacturing Controller, Production, International Edition: *Trudy Kimber*  
Cover Photo Credit: *kenoh / Shutterstock.com* <<http://Shutterstock.com>>  
Media Editor: *Daniel Sandin*.

---

Pearson Education Limited  
Edinburgh Gate  
Harlow  
Essex CM20 2JE  
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:  
[www.pearsoninternationaleditions.com](http://www.pearsoninternationaleditions.com)

© Pearson Education Limited 2013

The right of Paul Deitel and Harvey Deitel to be identified as authors of this work has been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

*Authorized adaptation from the United States edition, entitled C How to Program, 7<sup>th</sup> edition, ISBN 978-0-13-299044-8 by Paul Deitel and Harvey Deitel published by Pearson Education © 2013.*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6–10 Kirby Street, London EC1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

ISBN 10: 0-273-77684-3

ISBN 13: 978-0-273-77684-0

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

14 13 12 11 10

Typeset in AGaramond-Regular by GEX Publishing Services

Printed and bound by Courier Westford in The United States of America

The publisher's policy is to use paper manufactured from sustainable forests.

*In Memory of Dennis Ritchie,  
creator of the C programming language  
and co-creator of the UNIX operating system.*

*Paul and Harvey Deitel*

## Trademarks

MICROSOFT AND/OR ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE INFORMATION CONTAINED IN THE DOCUMENTS AND RELATED GRAPHICS PUBLISHED AS PART OF THE SERVICES FOR ANY PURPOSE. ALL SUCH DOCUMENTS AND RELATED GRAPHICS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT AND/OR ITS RESPECTIVE SUPPLIERS HEREBY DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS INFORMATION, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL MICROSOFT AND/OR ITS RESPECTIVE SUPPLIERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION AVAILABLE FROM THE SERVICES.

THE DOCUMENTS AND RELATED GRAPHICS CONTAINED HEREIN COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN. MICROSOFT AND/OR ITS RESPECTIVE SUPPLIERS MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED HEREIN AT ANY TIME. PARTIAL SCREEN SHOTS MAY BE VIEWED IN FULL WITHIN THE SOFTWARE VERSION SPECIFIED.

# Preface

Welcome to the C programming language—and to C++, too! This book presents leading-edge computing technologies for college students, instructors and software development professionals.

At the heart of the book is the Deitel signature “live-code approach.” We present concepts in the context of complete working programs, rather than in code snippets. Each code example is followed by one or more sample executions. Read the online Before You Begin section ([www.deitel.com/books/cht7/cht7\\_BYB.pdf](http://www.deitel.com/books/cht7/cht7_BYB.pdf)) to learn how to set up your computer to run the hundreds of code examples. All the source code is available at [www.deitel.com/books/cht7/](http://www.deitel.com/books/cht7/) and [www.pearsoninternationaleditions.com/deitel](http://www.pearsoninternationaleditions.com/deitel)<sup>①</sup>. Use the source code we provide to *run every program* as you study it.

We believe that this book and its support materials will give you an informative, challenging and entertaining introduction to C. As you read the book, if you have questions, send an e-mail to [deitel@deitel.com](mailto:deitel@deitel.com)—we’ll respond promptly. For book updates, visit [www.deitel.com/books/cht7/](http://www.deitel.com/books/cht7/), join our communities on Facebook ([www.deitel.com/deitelfan](http://www.deitel.com/deitelfan)), Twitter (@deitel) and Google+ ([gplus.to/deitel](http://gplus.to/deitel)), and subscribe to the *Deitel Buzz Online* newsletter ([www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)).

## New and Updated Features

Here are some key features of *C How to Program, 7/e*:

- **Coverage of the New C standard.** The previous edition of the book conformed to “standard C” and included a detailed appendix on the C99 standard. The New C Standard was approved just before *C How to Program, 7/e* went to publication. The new standard incorporates both C99 and the more recent C1X—now referred to as C11 or simply “the C standard” since its approval in 2011. Support for the new standard varies by compiler. The vast majority of our readership uses either the GNU gcc compiler—which supports several of the key features in the new standard—or the Microsoft Visual C++ compiler. Microsoft supports only a limited subset of the features that were added to C in C99 and C11—primarily the features that are also required by the C++ standard. To accommodate all of our readers, we placed the discussion of the new standard in optional, easy-to-use-or-omit sections and in Appendix F, Introduction to the New C Standard. We’ve also replaced various deprecated capabilities with newer preferred versions as a result of the new C standard.
- **New Chapter 1.** The new Chapter 1 engages students with intriguing facts and figures to get them excited about studying computers and computer programming. The chapter includes a table of some of the research made possible by computers and the Internet, current technology trends and hardware discussion, the data hierarchy, a new section on social networking, a table of business and technology publications and websites that will help you stay up to date with the latest technology news and trends, and updated exercises. We’ve included test-drives that show how to run a command-line C program on Microsoft Windows, Linux and Mac OS X.

---

① 也可登录华信教育资源网([www.hxedu.com.cn](http://www.hxedu.com.cn))下载本书相关代码。

- **Secure C Programming Sections.** We've added notes about secure C programming to many of the C programming chapters. We've also posted a Secure C Programming Resource Center at [www.deitel.com/SecureC/](http://www.deitel.com/SecureC/). For more details, see the section "A Note About Secure C Programming" in this Preface.
- **Focus on Performance Issues.** C (and C++) are favored by designers of performance-intensive applications such as operating systems, real-time systems, embedded systems and communications systems, so we focus intensively on performance issues.
- **"Making a Difference" Exercise Sets.** We encourage you to use computers and the Internet to research and solve problems that really matter. These exercises are meant to increase awareness of important issues the world is facing. We hope you'll approach them with your own values, politics and beliefs.
- **All Code Tested on Windows and Linux.** We've tested every example and exercise program using Visual C++ and GNU gcc in Windows and Linux, respectively.
- **Updated Coverage of C++ and Object-Oriented Programming.** We updated Chapters 15–24 on object-oriented programming in C++ with material from our textbook *C++ How to Program, 8/e*.
- **Sorting: A Deeper Look.** Sorting places data in order, based on one or more sort keys. We begin our presentation of sorting with a simple algorithm in Chapter 6—in Appendix E, we present a deeper look. We consider several algorithms and compare them with regard to their memory consumption and processor demands. For this purpose, we introduce Big O notation, which indicates how hard an algorithm may have to work to solve a problem. Through examples and exercises, Appendix E discusses the selection sort, insertion sort, recursive merge sort, recursive selection sort, bucket sort and recursive Quicksort. Sorting is an interesting problem because different sorting techniques achieve the same final result but they can vary hugely in their consumption of memory, CPU time and other system resources.
- **Titled Programming Exercises.** All the programming exercises are titled to help instructors tune assignments for their classes.
- **Debugger Appendices.** We've updated the Visual C++® and GNU gdb debugging appendices.
- **Order of Evaluation.** We added cautions about order of evaluation issues.
- **Additional Exercises.** We added more function pointer exercises. We also added a Fibonacci exercise project that improves the Fibonacci recursion example (tail recursion).
- **C++-Style // Comments.** We use the newer, more concise C++-style // comments in preference to C's older style /\*...\*/ comments.
- **C Standard Library.** Section 1.7 references P. J. Plauger's Dinkumware website ([www.dinkumware.com/manuals/default.aspx](http://www.dinkumware.com/manuals/default.aspx)) where students can find thorough searchable documentation for the C Standard Library functions.

## A Note About Secure C Programming

Throughout this book, we focus on C programming *fundamentals*. When we write each *How to Program* book, we search the corresponding language's standards document for the features that we feel novices need to learn in a first programming course, and features that existing programmers need to know to *begin* working in that language. We must also cover programming fundamentals *and* computer-science fundamentals for novice programmers—our core audience.

*Industrial-strength* coding techniques in any programming language are beyond the scope of an

introductory textbook. For that reason, our Secure C Programming sections present some key issues and techniques, and provide links and references so you can continue learning.

Experience has shown that it's difficult to build industrial-strength systems that stand up to attacks from viruses, worms, etc. Today, via the Internet, such attacks can be instantaneous and global in scope. Software vulnerabilities often come from simple programming issues. Building security into software from the start of the development cycle can greatly reduce costs and vulnerabilities.

The CERT<sup>®</sup> Coordination Center ([www.cert.org](http://www.cert.org)) was created to analyze and respond promptly to attacks. CERT—the Computer Emergency Response Team—publishes and promotes secure coding standards to help C programmers and others implement industrial-strength systems that avoid the programming practices that open systems to attacks. The CERT standards evolve as new security issues arise.

We've upgraded our code (as appropriate for an introductory book) to conform to various CERT recommendations. If you'll be building C systems in industry, consider reading *The CERT C Secure Coding Standard* (Robert Seacord, Addison-Wesley Professional, 2009) and *Secure Coding in C and C++* (Robert Seacord, Addison-Wesley Professional, 2006). The CERT guidelines are available free online at [www.securecoding.cert.org](http://www.securecoding.cert.org). Mr. Seacord, a technical reviewer for the C portion of this book, provided specific recommendations on each of our new Secure C Programming sections. Mr. Seacord is the Secure Coding Manager at CERT at Carnegie Mellon University's Software Engineering Institute (SEI) and an adjunct professor in the Carnegie Mellon University School of Computer Science.

The Secure C Programming sections at the ends of Chapters 2–13 discuss many important topics, including testing for arithmetic overflows, using unsigned integer types, new more secure functions in the C standard's Annex K, the importance of checking the status information returned by standard-library functions, range checking, secure random-number generation, array bounds checking, techniques for preventing buffer overflows, input validation, avoiding undefined behaviors, choosing functions that return status information vs. using similar functions that do not, ensuring that pointers are always NULL or contain valid addresses, using C functions vs. using preprocessor macros, and more.

## Web-Based Materials<sup>①</sup>

This book is supported by substantial online materials. The book's Companion Website ([www.pearsoninternationaleditions.com/deitel](http://www.pearsoninternationaleditions.com/deitel)) contains source code for all the code examples and the following appendices in *searchable* PDF format:

- Appendix E, Sorting: A Deeper Look
- Appendix F, Introduction to the New C Standard
- Appendix G, Using the Visual Studio Debugger
- Appendix H, Using the GNU Debugger

## Dependency Charts

Figures 1 and 2 show the dependencies among the chapters to help instructors plan their syllabi. *C How to Program*, 7/e is appropriate for CS1 and CS2 courses, and intermediate-level C and C++ programming courses. The C++ part of the book assumes that you've studied the C part.

---

① 也可登录华信教育资源网([www.hxedu.com.cn](http://www.hxedu.com.cn))下载相关资源。



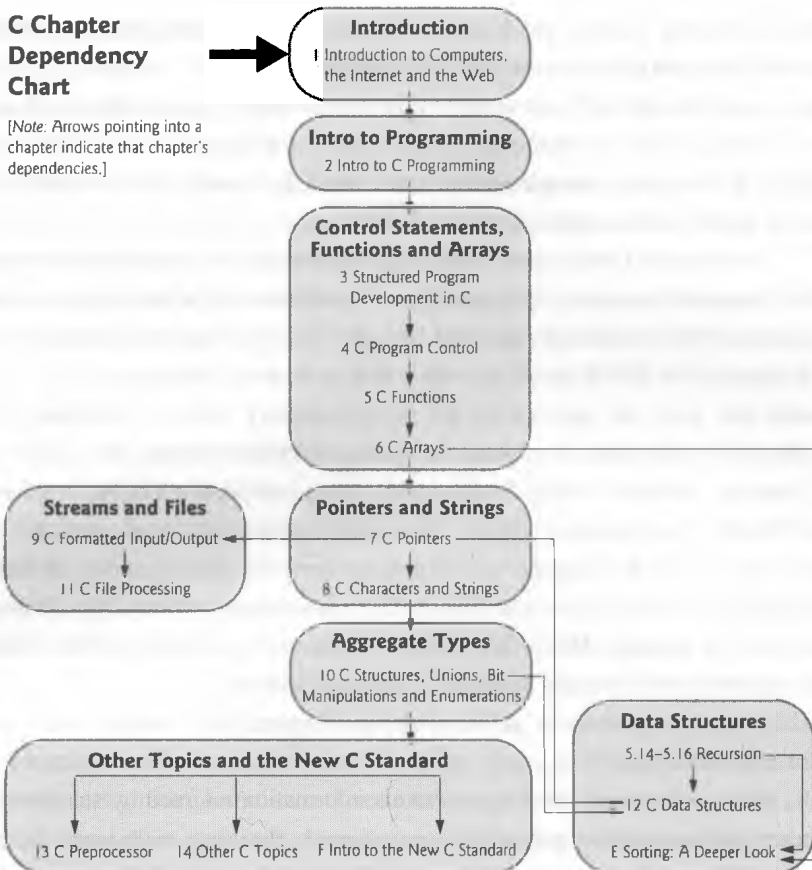


Fig. 1 | C chapter dependency chart.

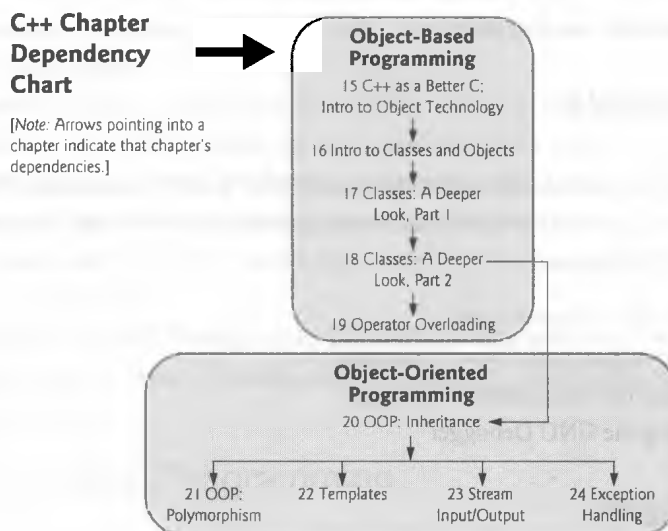


Fig. 2 | C++ chapter dependency chart.

## Teaching Approach

*C How to Program, 7/e*, contains a rich collection of examples. We focus on good software engineering and stressing program clarity.

**Syntax Shading.** For readability, we syntax shade the code, similar to the way most IDEs and code editors syntax color code. Our syntax-shading conventions are:

```
comments appear like this
keywords appear like this
constants and literal values appear like this
all other code appears in black
```

**Code Highlighting.** We place gray rectangles around the key code.

**Using Fonts for Emphasis.** We place the key terms and the index's page reference for each defining occurrence in bold text for easy reference. We emphasize on-screen components in the bold Helvetica font (e.g., the File menu) and C program text in the Lucida font (for example, `int x = 5;`).

**Objectives.** The opening quotes are followed by a list of chapter objectives.

**Illustrations/Figures.** Abundant charts, tables, line drawings, UML diagrams, programs and program output are included.

**Programming Tips.** We include programming tips to help you focus on important aspects of program development. These tips and practices represent the best we've gleaned from a combined seven decades of programming and teaching experience.



#### Good Programming Practice

*The Good Programming Practices call attention to techniques that will help you produce programs that are clearer, more understandable and more maintainable.*



#### Common Programming Error

*Pointing out these Common Programming Errors reduces the likelihood that you'll make them.*



#### Error-Prevention Tip

*These tips contain suggestions for exposing and removing bugs from your programs; many describe aspects of C that prevent bugs from getting into programs in the first place.*



#### Performance Tip

*These tips highlight opportunities for making your programs run faster or minimizing the amount of memory that they occupy.*



#### Portability Tip

*The Portability Tips help you write code that will run on a variety of platforms.*



#### Software Engineering Observation

*The Software Engineering Observations highlight architectural and design issues that affect the construction of software systems, especially large-scale systems.*

**Summary Bullets.** We present a section-by-section, bullet-list summary of the chapter.

**Terminology.** We include an alphabetized list of the important terms defined in each chapter.

**Self-Review Exercises and Answers.** Extensive self-review exercises and answers are included for self-study.

**Exercises.** Each chapter concludes with a substantial set of exercises including:

- simple recall of important terminology and concepts
- identifying the errors in code samples
- writing individual program statements

- writing small portions of C functions and C++ member functions and classes
- writing complete programs
- implementing major projects

**Index.** We've included an extensive index, which is especially useful when you use the book as a reference.

## Software Used in C How to Program, 7/e

We wrote *C How to Program, 7/e* using Microsoft's free Visual C++ Express Edition (which can compile both C and C++ programs and can be downloaded from [www.microsoft.com/express/downloads/](http://www.microsoft.com/express/downloads/)) and the free GNU C and C++ compilers ([gcc.gnu.org/install/binaries.html](http://gcc.gnu.org/install/binaries.html)), which are already installed on most Linux systems and can be installed on Mac OS X and Windows systems. Apple includes GNU C and C++ in their Xcode development tools, which Mac OS X users can download from [developer.apple.com/technologies/tools/xcode.html](http://developer.apple.com/technologies/tools/xcode.html).

For other free C and C++ compilers, visit:

[www.thefreecountry.com/compilers/cpp.shtml](http://www.thefreecountry.com/compilers/cpp.shtml)  
[www.compilers.net/Dir/Compilers/CCpp.htm](http://www.compilers.net/Dir/Compilers/CCpp.htm)  
[www.freebyte.com/programming/cpp/#cppcompilers](http://www.freebyte.com/programming/cpp/#cppcompilers)  
[en.wikipedia.org/wiki/List\\_of\\_compilers#C.2B.2B\\_compilers](http://en.wikipedia.org/wiki/List_of_compilers#C.2B.2B_compilers)

## CourseSmart Web Books

Today's students and instructors have increasing demands on their time and money. Pearson has responded to that need by offering digital texts and course materials online through CourseSmart. CourseSmart allows faculty to review course materials online, saving time and costs. It offers students a high-quality digital version of the text for less than the cost of a print copy. Students receive the same content offered in the print textbook enhanced by search, note-taking and printing tools. For more information, visit

[www.coursesmart.co.uk](http://www.coursesmart.co.uk)

## Instructor Resources<sup>①</sup>

The following supplements are available to *qualified instructors only* through Pearson Education's Instructor Resource Center [www.pearsoninternationaleditions.com/deitel](http://www.pearsoninternationaleditions.com/deitel).

- **PowerPoint® slides** containing all the code and figures in the text, plus bulleted items that summarize key points.
- **Test Item File** of multiple-choice questions (approximately two per book section)
- **Solutions Manual** with solutions to most of the end-of-chapter exercises. Please check the Instructor Resource Center to determine which exercises have solutions.

Please do not write to us requesting access to the Pearson Instructor's Resource Center. Access is restricted to college instructors teaching from the book. Instructors may obtain access only through their Pearson representatives. If you're not a registered faculty member, contact your Pearson representative.

Solutions are *not* provided for "project" exercises. Check out our Programming Projects Resource Center for lots of additional exercise and project possibilities ([www.deitel.com/ProgrammingProjects/](http://www.deitel.com/ProgrammingProjects/)).

---

① 教辅资源申请方式请参见书末的“教学支持说明”。

# Acknowledgments

We'd like to thank Abbey Deitel and Barbara Deitel for long hours devoted to this project. We're fortunate to have worked with the dedicated team of publishing professionals at Pearson. We appreciate the guidance, savvy and energy of Michael Hirsch, Editor-in-Chief of Computer Science. Carole Snyder and Bob Engelhardt did a marvelous job managing the review and production processes, respectively.

## **C How to Program, 7/e Reviewers**

We wish to acknowledge the efforts of our reviewers. Under tight deadlines, they scrutinized the text and the programs and provided countless suggestions for improving the presentation: Dr. John F. Doyle (Indiana University Southeast), Hemanth H.M. (Software Engineer at SonicWALL), Vytautus Leonavicius (Microsoft), Robert Seacord (Secure Coding Manager at SEI/CERT, author of *The CERT C Secure Coding Standard* and technical expert for the international standardization working group for the programming language C) and José Antonio González Seco (Parliament of Andalusia).

## **Other Recent Editions Reviewers**

William Albrecht (University of South Florida), Ian Barland (Radford University), Ed James Beckham (Altera), John Benito (Blue Pilot Consulting, Inc. and Convener of ISO WG14—the Working Group responsible for the C Programming Language Standard), Alireza Fazelpour (Palm Beach Community College), Mahesh Hariharan (Microsoft), Kevin Mark Jones (Hewlett Packard), Lawrence Jones, (UGS Corp.), Don Kostuch (Independent Consultant), Xiaolong Li (Indiana State University), William Mike Miller (Edison Design Group, Inc.), Tom Rethard (The University of Texas at Arlington), Benjamin Seyfarth (University of Southern Mississippi), Gary Sibbitts (St. Louis Community College at Meramec), William Smith (Tulsa Community College) and Douglas Walls (Senior Staff Engineer, C compiler, Sun Microsystems).

Well, there you have it! C is a powerful programming language that will help you write high-performance programs quickly and effectively. C scales nicely into the realm of enterprise systems development to help organizations build their business-critical and mission-critical information systems. As you read the book, we would sincerely appreciate your comments, criticisms, corrections and suggestions for improving the text. Please address all correspondence to:

deitel@deitel.com

We'll respond promptly, and post corrections and clarifications on:

[www.deitel.com/books/cht7/](http://www.deitel.com/books/cht7/)

We hope you enjoy working with *C How to Program, Seventh Edition* as much as we enjoyed writing it!

*Paul Deitel*

*Harvey Deitel*

January 2012

The publishers wish to thank Atriya Sen, of the Indian Statistical Institute, Kolkata, for reviewing the content of the International Edition.

# About the Authors

**Paul Deitel**, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of MIT, where he studied Information Technology. Through Deitel & Associates, Inc., he has delivered hundreds of programming courses to industry clients, including Cisco, IBM, Siemens, Sun Microsystems, Dell, Lucent Technologies, Fidelity, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, White Sands Missile

Range, Rogue Wave Software, Boeing, SunGard Higher Education, Stratus, Cambridge Technology Partners, One Wave, Hyperion Software, Adra Systems, Entergy, CableData Systems, Nortel Networks, Puma, iRobot, Invensys and many more. He and his co-author, Dr. Harvey M. Deitel, are the world's best-selling programming-language textbook/professional book/video authors.

**Dr. Harvey Deitel**, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 50 years of experience in the computer field. Dr. Deitel earned B.S. and M.S. degrees from MIT and a Ph.D. from Boston University. He has extensive college teaching experience, including earning tenure and serving as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc., in 1991 with his son, Paul Deitel. The Deitels' publications have earned international recognition, with translations published in Chinese, Korean, Japanese, German, Russian, Spanish, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish. Dr. Deitel has delivered hundreds of professional programming seminars to major corporations, academic institutions, government organizations and the military.

## Corporate Training from Deitel & Associates, Inc.

Deitel & Associates, Inc., founded by Paul Deitel and Harvey Deitel, is an internationally recognized authoring, corporate training and software development organization specializing in computer programming languages, object technology, Android and iPhone app development and Internet and web software technology. The company offers instructor-led training courses delivered at client sites worldwide on major programming languages and platforms, including C, C++, Visual C++®, Java, Visual C#®, Visual Basic®, XML®, Python®, object technology, Internet and web programming, Android app development, Objective-C and iPhone app development and a growing list of additional programming and software development courses. The company's clients include many of the world's largest companies, government agencies, branches of the military, and academic institutions.

Through its 36-year publishing partnership with Prentice Hall/Pearson, Deitel & Associates, Inc., publishes leading-edge programming college textbooks, professional books and *LiveLessons* video courses. Deitel & Associates, Inc. and the authors can be reached at:

[deitel@deitel.com](mailto:deitel@deitel.com)

To learn more about Deitel's *Dive Into Series* Corporate Training curriculum, visit:

[www.deitel.com/training/](http://www.deitel.com/training/)

To request a proposal for worldwide on-site, instructor-led training at your company or organization, e-mail [deitel@deitel.com](mailto:deitel@deitel.com).

Individuals wishing to purchase Deitel books and *LiveLessons* video training can do so through [www.deitel.com](http://www.deitel.com). Bulk orders by corporations, the government, the military and academic institutions should be placed directly with Pearson.

# Contents

<b>Chapter 1</b>	<b>Introduction to Computers, the Internet and the Web</b>	<b>1</b>
1.1	Introduction	2
1.2	Computers and the Internet in Industry and Research	2
1.3	Hardware and Software	4
1.4	Data Hierarchy	6
1.5	Programming Languages	7
1.6	The C Programming Language	7
1.7	C Standard Library	9
1.8	C++ and Other C-Based Languages	9
1.9	Object Technology	10
1.10	Typical C Program Development Environment	12
1.11	Test-Driving a C Application in Windows, Linux and Mac OS X	14
1.12	Operating Systems	21
1.13	The Internet and World Wide Web	23
1.14	Some Key Software Development Terminology	24
1.15	Keeping Up-to-Date with Information Technologies	25
1.16	Web Resources	26
<b>Chapter 2</b>	<b>Introduction to C Programming</b>	<b>32</b>
2.1	Introduction	32
2.2	A Simple C Program: Printing a Line of Text	32
2.3	Another Simple C Program: Adding Two Integers	36
2.4	Memory Concepts	39
2.5	Arithmetic in C	40
2.6	Decision Making: Equality and Relational Operators	43
2.7	Secure C Programming	46
<b>Chapter 3</b>	<b>Structured Program Development in C</b>	<b>58</b>
3.1	Introduction	58
3.2	Algorithms	59
3.3	Pseudocode	59
3.4	Control Structures	59
3.5	The if Selection Statement	61
3.6	The if-else Selection Statement	62
3.7	The while Repetition Statement	65
3.8	Formulating Algorithms Case Study 1: CounterControlled Repetition	66
3.9	Formulating Algorithms with Top-Down, Stepwise Refinement Case Study 2: Sentinel-Controlled Repetition	67
3.10	Formulating Algorithms with Top-Down, Stepwise Refinement Case Study 3: Nested Control Statements	72

3.11	Assignment Operators	75
3.12	Increment and Decrement Operators	76
3.13	Secure C Programming	78
<b>Chapter 4</b>	<b>C Program Control</b>	<b>95</b>
4.1	Introduction	95
4.2	Repetition Essentials	96
4.3	Counter-Controlled Repetition	96
4.4	for Repetition Statement	98
4.5	for Statement: Notes and Observations	100
4.6	Examples Using the for Statement	100
4.7	switch Multiple-Selection Statement	103
4.8	do-while Repetition Statement	108
4.9	break and continue Statements	109
4.10	Logical Operators	111
4.11	Confusing Equality (==) and Assignment (=) Operators	113
4.12	Structured Programming Summary	114
4.13	Secure C Programming	118
<b>Chapter 5</b>	<b>C Functions</b>	<b>132</b>
5.1	Introduction	133
5.2	Program Modules in C	133
5.3	Math Library Functions	134
5.4	Functions	135
5.5	Function Definitions	135
5.6	Function Prototypes: A Deeper Look	139
5.7	Function Call Stack and Stack Frames	141
5.8	Headers	143
5.9	Passing Arguments By Value and By Reference	144
5.10	Random Number Generation	145
5.11	Example: A Game of Chance	148
5.12	Storage Classes	151
5.13	Scope Rules	152
5.14	Recursion	155
5.15	Example Using Recursion: Fibonacci Series	158
5.16	Recursion vs. Iteration	160
5.17	Secure C Programming	162
<b>Chapter 6</b>	<b>C Arrays</b>	<b>181</b>
6.1	Introduction	181
6.2	Arrays	182
6.3	Defining Arrays	182
6.4	Array Examples	183
6.5	Passing Arrays to Functions	193
6.6	Sorting Arrays	196

6.7	Case Study: Computing Mean, Median and Mode Using Arrays .....	198
6.8	Searching Arrays.....	202
6.9	Multidimensional Arrays.....	206
6.10	Variable-Length Arrays.....	211
6.11	Secure C Programming .....	213
<b>Chapter 7</b>	<b>C Pointers .....</b>	<b>230</b>
7.1	Introduction .....	231
7.2	Pointer Variable Definitions and Initialization .....	231
7.3	Pointer Operators .....	232
7.4	Passing Arguments to Functions by Reference .....	233
7.5	Using the const Qualifier with Pointers .....	235
7.6	Bubble Sort Using Pass-by-Reference.....	241
7.7	sizeof Operator.....	243
7.8	Pointer Expressions and Pointer Arithmetic.....	245
7.9	Relationship between Pointers and Arrays.....	247
7.10	Arrays of Pointers.....	250
7.11	Case Study: Card Shuffling and Dealing Simulation.....	251
7.12	Pointers to Functions .....	254
7.13	Secure C Programming .....	258
<b>Chapter 8</b>	<b>C Characters and Strings .....</b>	<b>277</b>
8.1	Introduction .....	278
8.2	Fundamentals of Strings and Characters.....	278
8.3	Character-Handling Library .....	280
8.4	String-Conversion Functions.....	284
8.5	Standard Input/Output Library Functions .....	286
8.6	String-Manipulation Functions of the StringHandling Library.....	289
8.7	Comparison Functions of the String-Handling Library .....	291
8.8	Search Functions of the String-Handling Library.....	292
8.9	Memory Functions of the String-Handling Library.....	297
8.10	Other Functions of the String-Handling Library .....	300
8.11	Secure C Programming .....	301
<b>Chapter 9</b>	<b>C Formatted Input/Output .....</b>	<b>314</b>
9.1	Introduction .....	314
9.2	Streams.....	315
9.3	Formatting Output with printf.....	315
9.4	Printing Integers .....	315
9.5	Printing Floating-Point Numbers.....	316
9.6	Printing Strings and Characters .....	318
9.7	Other Conversion Specifiers .....	318
9.8	Printing with Field Widths and Precision.....	319
9.9	Using Flags in the printf Format Control String .....	321
9.10	Printing Literals and Escape Sequences.....	323



9.11	Reading Formatted Input with <code>scanf</code> .....	323
9.12	Secure C Programming .....	328
<b>Chapter 10</b>	<b>C Structures, Unions, Bit Manipulation and Enumerations .....</b>	<b>335</b>
10.1	Introduction .....	336
10.2	Structure Definitions .....	336
10.3	Initializing Structures .....	338
10.4	Accessing Structure Members .....	338
10.5	Using Structures with Functions .....	340
10.6	<code>typedef</code> .....	340
10.7	Example: High-Performance Card Shuffling and Dealing Simulation .....	341
10.8	Unions .....	343
10.9	Bitwise Operators .....	344
10.10	Bit Fields .....	351
10.11	Enumeration Constants .....	353
10.12	Secure C Programming .....	355
<b>Chapter 11</b>	<b>C File Processing .....</b>	<b>365</b>
11.1	Introduction .....	365
11.2	Files and Streams .....	366
11.3	Creating a Sequential-Access File .....	366
11.4	Reading Data from a Sequential-Access File .....	370
11.5	Random-Access Files .....	373
11.6	Creating a Random-Access File .....	374
11.7	Writing Data Randomly to a Random-Access File .....	375
11.8	Reading Data from a Random-Access File .....	377
11.9	Case Study: Transaction-Processing Program .....	379
11.10	Secure C Programming .....	383
<b>Chapter 12</b>	<b>C Data Structures .....</b>	<b>393</b>
12.1	Introduction .....	394
12.2	Self-Referential Structures .....	394
12.3	Dynamic Memory Allocation .....	395
12.4	Linked Lists .....	396
12.5	Stacks .....	402
12.6	Queues .....	406
12.7	Trees .....	411
12.8	Secure C Programming .....	415
<b>Chapter 13</b>	<b>C Preprocessor .....</b>	<b>427</b>
13.1	Introduction .....	427
13.2	<code>#include</code> Preprocessor Directive .....	428
13.3	<code>#define</code> Preprocessor Directive: Symbolic Constants .....	428
13.4	<code>#define</code> Preprocessor Directive: Macros .....	429
13.5	Conditional Compilation .....	430
13.6	<code>#error</code> and <code>#pragma</code> Preprocessor Directives .....	431

13.7	# and ## Operators	432
13.8	Line Numbers	432
13.9	Predefined Symbolic Constants	432
13.10	Assertions	433
13.11	Secure C Programming	433
Chapter 14	Other C Topics	438
14.1	Introduction	438
14.2	Redirecting I/O	438
14.3	Variable-Length Argument Lists	439
14.4	Using Command-Line Arguments	441
14.5	Notes on Compiling Multiple-Source-File Programs	442
14.6	Program Termination with <code>exit</code> and <code>atexit</code>	443
14.7	Suffixes for Integer and Floating-Point Literals	444
14.8	Signal Handling	445
14.9	Dynamic Memory Allocation: Functions <code>calloc</code> and <code>realloc</code>	447
14.10	Unconditional Branching with <code>goto</code>	447
Chapter 15	C++ as a Better C; Introducing Object Technology	453
15.1	Introduction	454
15.2	C++	454
15.3	A Simple Program: Adding Two Integers	454
15.4	C++ Standard Library	456
15.5	Header Files	457
15.6	Inline Functions	458
15.7	References and Reference Parameters	460
15.8	Empty Parameter Lists	464
15.9	Default Arguments	464
15.10	Unary Scope Resolution Operator	466
15.11	Function Overloading	467
15.12	Function Templates	469
15.13	Introduction to C++ Standard Library Class Template <code>vector</code>	471
15.14	Introduction to Object Technology and the UML	476
15.15	Wrap-Up	479
Chapter 16	Introduction to Classes, Objects and Strings	486
16.1	Introduction	486
16.2	Defining a Class with a Member Function	487
16.3	Defining a Member Function with a Parameter	489
16.4	Data Members, <i>set</i> Functions and <i>get</i> Functions	492
16.5	Initializing Objects with Constructors	496
16.6	Placing a Class in a Separate File for Reusability	499
16.7	Separating Interface from Implementation	502
16.8	Validating Data with <i>set</i> Functions	507
16.9	Wrap-Up	510

<b>Chapter 17</b>	<b>Classes: A Deeper Look, Part 1</b>	517
17.1	Introduction	517
17.2	Time Class Case Study	518
17.3	Class Scope and Accessing Class Members	524
17.4	Separating Interface from Implementation	525
17.5	Access Functions and Utility Functions	526
17.6	Time Class Case Study: Constructors with Default Arguments	528
17.7	Destructors	532
17.8	When Constructors and Destructors Are Called	532
17.9	Time Class Case Study: A Subtle Trap—Returning a Reference to a private Data Member	535
17.10	Default Memberwise Assignment	537
17.11	Wrap-Up	539
<b>Chapter 18</b>	<b>Classes: A Deeper Look, Part 2</b>	545
18.1	Introduction	545
18.2	const (Constant) Objects and const Member Functions	546
18.3	Composition: Objects as Members of Classes	551
18.4	friend Functions and friend Classes	556
18.5	Using the this Pointer	558
18.6	static Class Members	562
18.7	Proxy Classes	565
18.8	Wrap-Up	568
<b>Chapter 19</b>	<b>Operator Overloading; Class string</b>	574
19.1	Introduction	574
19.2	Using the Overloaded Operators of Standard Library Class string	575
19.3	Fundamentals of Operator Overloading	578
19.4	Overloading Binary Operators	579
19.5	Overloading the Binary Stream Insertion and Stream Extraction Operators	579
19.6	Overloading Unary Operators	583
19.7	Overloading the Unary Prefix and Postfix ++ and -- Operators	583
19.8	Case Study: A Date Class	584
19.9	Dynamic Memory Management	588
19.10	Case Study: Array Class	590
19.11	Operators as Member Functions vs. Non-Member Functions	599
19.12	Converting between Types	600
19.13	explicit Constructors	601
19.14	Building a String Class	603
19.15	Wrap-Up	603
<b>Chapter 20</b>	<b>Object-Oriented Programming: Inheritance</b>	614
20.1	Introduction	615
20.2	Base Classes and Derived Classes	615
20.3	protected Members	617
20.4	Relationship between Base Classes and Derived Classes	617

20.5	Constructors and Destructors in Derived Classes	635
20.6	public, protected and private Inheritance	636
20.7	Software Engineering with Inheritance	636
20.8	Wrap-Up	637
<b>Chapter 21</b>	<b>Object-Oriented Programming: Polymorphism</b>	<b>642</b>
21.1	Introduction	643
21.2	Introduction to Polymorphism: Polymorphic Video Game	643
21.3	Relationships Among Objects in an Inheritance Hierarchy	644
21.4	Type Fields and switch Statements	653
21.5	Abstract Classes and Pure virtual Functions	654
21.6	Case Study: Payroll System Using Polymorphism	656
21.7	(Optional) Polymorphism, Virtual Functions and Dynamic Binding “Under the Hood”	667
21.8	Case Study: Payroll System Using Polymorphism and Runtime Type Information with Downcasting, dynamic_cast, typeid and type_info	670
21.9	Virtual Destructors	672
21.10	Wrap-Up	673
<b>Chapter 22</b>	<b>Templates</b>	<b>679</b>
22.1	Introduction	679
22.2	Function Templates	680
22.3	Overloading Function Templates	682
22.4	Class Templates	683
22.5	Nontype Parameters and Default Types for Class Templates	687
22.6	Wrap-Up	688
<b>Chapter 23</b>	<b>Stream Input/Output</b>	<b>692</b>
23.1	Introduction	693
23.2	Streams	693
23.3	Stream Output	696
23.4	Stream Input	697
23.5	Unformatted I/O Using read, write and gcount	700
23.6	Introduction to Stream Manipulators	701
23.7	Stream Format States and Stream Manipulators	705
23.8	Stream Error States	712
23.9	Tying an Output Stream to an Input Stream	714
23.10	Wrap-Up	714
<b>Chapter 24</b>	<b>Exception Handling: A Deeper Look</b>	<b>723</b>
24.1	Introduction	723
24.2	Example: Handling an Attempt to Divide by Zero	724
24.3	When to Use Exception Handling	728
24.4	Rethrowing an Exception	729
24.5	Processing Unexpected Exceptions	731
24.6	Stack Unwinding	731
24.7	Constructors, Destructors and Exception Handling	732

24.8 Exceptions and Inheritance ..... 733

24.9 Processing new Failures ..... 733

24.10 Class unique\_ptr and Dynamic Memory Allocation ..... 735

24.11 Standard Library Exception Hierarchy ..... 737

24.12 Wrap-Up ..... 739

Appendix A Operator Precedence Charts ..... 744

Appendix B ASCII Character Set ..... 747

Appendix C Number Systems ..... 748

Appendix D Game Programming: Solving Sudoku ..... 759

Index ..... 766

附录 E ~ 附录 H 请登录华信教育资源网 (www.hxedu.com.cn) 注册下载。

Appendix E Sorting: A Deeper Look

Appendix F Introduction to the New C Standard

Appendix G Using the Visual Studio Debugger

Appendix H Using the GNU Debugger