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/chtp7/chtp7_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/chtp7/` and `www.pearsonhighered.com/deitel`. 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`—we'll respond promptly. For book updates, visit `www.deitel.com/books/chtp7/`, join our communities on Facebook (`www.deitel.com/deitelfan`), Twitter (`@deitel`) and Google+ (`gplus.to/deitel`), and subscribe to the *Deitel® Buzz Online* newsletter (`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 pub-

lications 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.

- ***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/`. 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`) 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® Coordination Center (`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`. 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

This book is supported by substantial online materials. The book's Companion Website (`www.pearsonhighered.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.

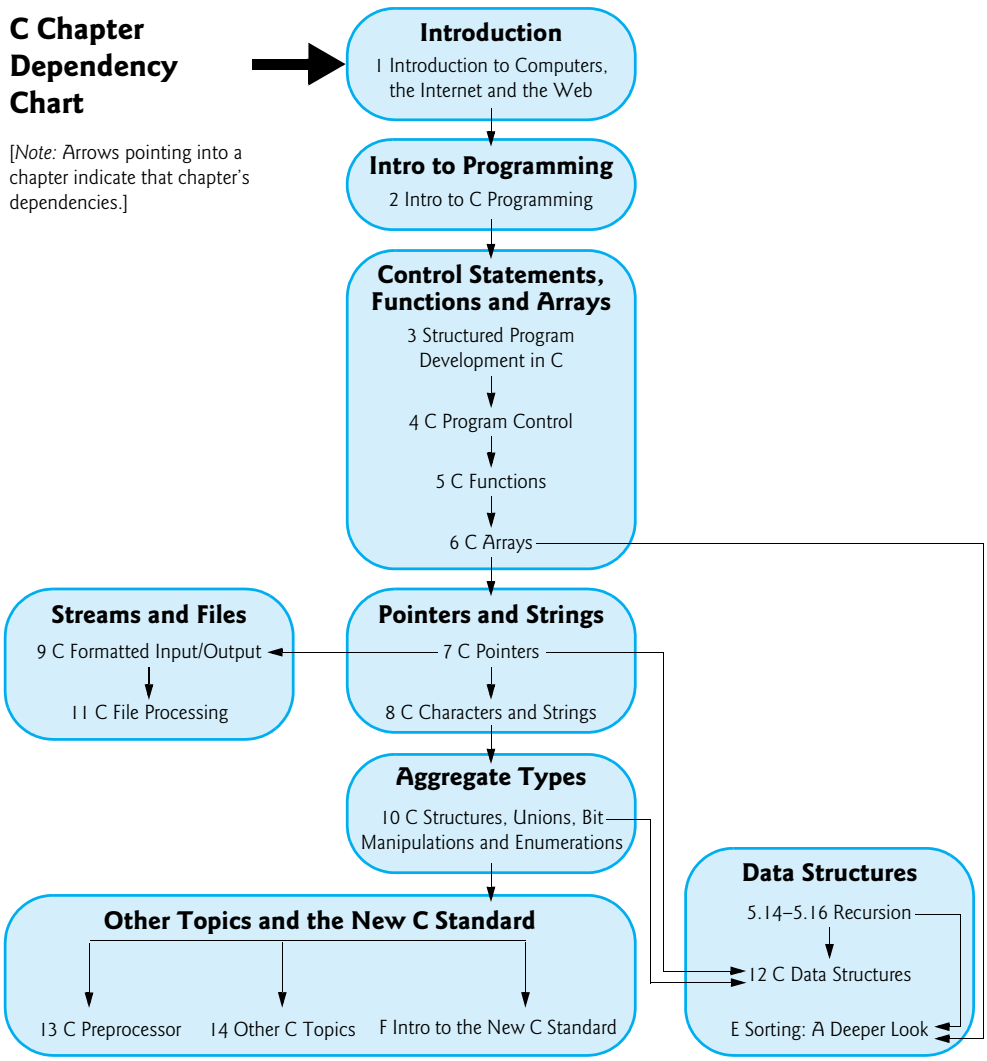

Fig. 1 | C chapter dependency chart.

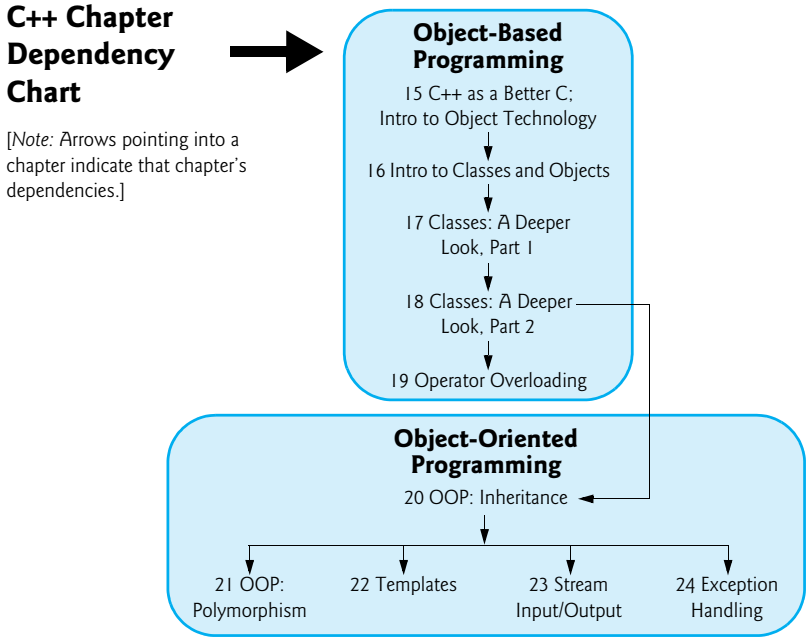

Fig. 2 | C++ chapter dependency chart.

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 blue** 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 Practices

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

Common Programming Errors

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

Error-Prevention Tips

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 Tips

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

Portability Tips

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

Software Engineering Observations

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 with the page number of each term's defining occurrence for easy reference.

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. Defining occurrences of key terms are highlighted with a **bold blue** page number.

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/) and the free GNU C and C++ compilers (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.

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

```
www.thefreecountry.com/compilers/cpp.shtml
www.compilers.net/Dir/Compilers/CCpp.htm
www.freebyte.com/programming/cpp/#cppcompilers
en.wikipedia.org/wiki/List_of_compilers#C.2B.2B_compilers
```

C++ IDE Resource Kit

Your instructor may have ordered through your college bookstore a Value Pack edition of *C How to Program, 7/e* that comes bundled with the C++ IDE Resource Kit—most C++ compilers also support C. This kit contains CD or DVD versions of:

- Microsoft® Visual Studio 2010 Express Edition (www.microsoft.com/express/)
- Dev C++ (www.bloodshed.net/download.html)
- NetBeans (netbeans.org/downloads/index.html)
- Eclipse (eclipse.org/downloads/)
- CodeLite (codelite.org/LiteEditor/Download)

You can also download these software packages from the websites specified above. The C++ IDE Resource Kit also includes access to a Companion Website containing step-by-step written instructions and VideoNotes to help you get started with each development environment. If your book did not come with the C++ IDE Resource Kit, you can purchase access to the Resource Kit's Companion Website from www.pearsonhighered.com/cppidekit/.

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.com.

Instructor Resources

The following supplements are available to *qualified instructors only* through Pearson Education's Instructor Resource Center (www.pearsonhighered.com/irc):

- ***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 or visit `www.pearsonhighered.com/educator/replocator/`.

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/`).

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/chtp7/
```

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

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
```

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

```
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`.

Individuals wishing to purchase Deitel books and *LiveLessons* video training can do so through `www.deitel.com`. Bulk orders by corporations, the government, the military and academic institutions should be placed directly with Pearson. For more information, visit `www.pearsoned.com/professional/index.htm`.