



Preface

“The chief merit of language is clearness ...”

—Galen

Welcome to the world of C++ programming and *C++ How to Program, Seventh Edition!* This book presents leading-edge computing technologies for students, instructors and software development professionals.

At the heart of the book is the Deitel signature “live-code approach.” Concepts are presented in the context of complete working C++ programs, rather than in code snippets. Each code example is immediately followed by one or more sample executions. All the source code is available at www.deitel.com/books/cpphtp7/.

New and Updated Features

Here are the updates we’ve made for *C++ How to Program, 7/e*:

- **“Making a Difference” Exercise Sets.** We encourage you to use computers and the Internet to research and solve problems that really matter. These new 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.
- **Prefer *string* Objects to C Strings.** C++ offers two types of strings—string class objects (which we use starting in Chapter 3) and C-style, pointer-based strings. We continue to include some early discussions of C strings to give you practice with pointer manipulations, to illustrate dynamic memory allocation with new and delete and to prepare you for working with C strings in the “legacy code” that you’ll encounter in industry. In new development, you should favor string class objects. We’ve replaced most occurrences of C strings with instances of C++ class string to make programs more robust and eliminate many of the security problems that can be caused by manipulating C strings.
- **Prefer *vectors* to C Arrays.** Similarly, C++ offers two types of arrays—vector class objects (which we use starting in Chapter 7) and C-style, pointer-based arrays. As appropriate, we use class template vector instead of C arrays throughout the book. However, we begin by discussing C arrays in Chapter 7 to prepare you for working with legacy code and to use as a basis for building your own customized Array class in Chapter 11, Operator Overloading.
- **New Companion Website (www.pearsonhighered.com/deitel/).** This edition’s Companion Website includes a wealth of material to help you with your study of C++ programming. We provide an extensive number of VideoNotes that walk you through the code examples in 14 of the key chapters, solutions to many of the book’s exercises, bonus chapters, and more (see the Companion Website section later in this Preface).

- *Dynamic Memory Allocation.* We moved dynamic memory allocation later in the book to Chapter 11, where it's first needed. The "proxy class" discussion (which uses dynamic memory) has also been moved to Chapter 11.
- *Titled Programming Exercises.* We've titled all the programming exercises. This and the previous two features help instructors tune assignments for their classes.
- *Eliminated "Magic" Numbers.* We eliminated all uses of truly "magic" numbers and replaced them with named constants or enums as appropriate. In a few cases in which the context is absolutely clear, we don't consider numbers to be "magic."
- *Enhanced Use of `const`.* We increased our use of `const` bookwide to encourage better software engineering.
- *Eliminated "return 0;".* According to the C++ standard, any `main` function that does not contain "return 0;" as its last statement is assumed to return 0. For this reason, we've eliminated "return 0;" from all but the first program in the book.
- *Use "using namespace std;".* Previously, we specified a using declaration for every individual item that we referenced from a C++ Standard Library header file. Since these items are well known and unlikely to have name collisions with other C++ libraries, we now use "using namespace std;" for all C++ Standard Library components from Chapter 3 forward. This simplifies the programs and saves many lines of code.
- *New Design.* The book has a new interior design that graphically serves to organize, clarify and highlight the information, and enhances the book's pedagogy.
- *Reorganized Optional OOD Case Study.* We tuned the Object-Oriented Design/UML automated teller machine (ATM) case study and reorganized it into two optional chapters (25 and 26) that present the ATM's design and complete code implementation. This is a nice business example that most students can relate to. Working through these two chapters as a unit will help you tie together many of the object-oriented programming (OOP) concepts you learn in Chapters 1–13. A key concept in OOP is the interactions among objects. In most textbooks, the code examples create and use only one or two objects. The ATM case study gives you the opportunity to examine the interactions among *many* objects that provide the functionality of a substantial system. For instructors who wish to cover the case study in a distributed manner, we've indicated where each section in Chapters 25 and 26 can be covered inline with earlier chapters in the book.
- *Function Pointer Exercises.* We added several real-world function-pointers exercises. These are available at the Companion Website and at www.deitel.com/books/cpphttp7/.
- *Improved Terminology Sections.* We've added page numbers for the defining occurrences of all terms in the terminology lists for easy reference.

New Features in the Next C++ Standard

We discuss four new language features that will be part of the next C++ standard and are already implemented by some of today's C++ compilers. These include:

- *Initializer Lists for User-Defined Types.* These enable objects of your own types to be initialized using the same syntax as built-in arrays.

- **Range-Based for Statement.** A version of the for statement that iterates over all the elements of an array or container (such as an object of the vector class).
- **Lambda Expressions.** These enable you to create anonymous functions that can be passed to other functions as arguments.
- **Concepts.** These enable template programmers to specify the requirements for data types that will be used with a particular template. Compilers can then provide more meaningful error messages when data types do not meet a template's requirements.

Other Features

Other features of *C++ How to Program, 7/e*, include:

- **Game Programming.** The computer-game industry's revenues are already greater than those of the first-run movie business, creating lots of career opportunities. Chapter 27, Game Programming with Ogre, introduces game programming and graphics with the open source Ogre 3D graphics engine. We discuss basic issues involved in game programming. Then we show how to use Ogre to create a simple game featuring a play mechanic similar to the classic video game Pong[®], originally developed by Atari. We demonstrate how to create a scene with 3D color graphics, smoothly animate moving objects, use timers to control animation speed, detect collisions between objects, add sound, accept keyboard input and display text output.
- **Future of C++.** Chapter 23 considers the future of C++—we introduce the Boost C++ Libraries, Technical Report 1 (TR1) and C++0x. The free Boost open source libraries are created by members of the C++ community. Technical Report 1 describes the proposed changes to the C++ Standard Library, many of which are based on current Boost libraries. The C++ Standards Committee is revising the C++ Standard. The main goals for the new standard are to make C++ easier to learn, improve library building capabilities, and increase compatibility with the C programming language. The last standard was published in 1998. The new standard is likely to be released in 2010 or 2011. It will include changes to the core language and many of the libraries in TR1. We overview the Boost libraries and provide code examples for the “regular expression” and “smart pointer” libraries. Regular expressions are used to match specific character patterns in text. They can be used, for example, to validate data to ensure that it's in a particular format, to replace parts of one string with another, or to split a string. Many common bugs in C and C++ code are related to pointers, a powerful programming capability you'll study in Chapter 8, Pointers. Smart pointers help you avoid errors by providing additional functionality to standard pointers.
- **Integrated Case Studies.** We provide several case studies spanning multiple sections and chapters. These include the development of the GradeBook class in Chapters 3–7, the Time class in Chapters 9–10, the Employee class in Chapters 12–13, and the optional OOD/UML ATM case study in Chapters 25–26.
- **Integrated GradeBook Case Study.** The GradeBook case study uses classes and objects in Chapters 3–7 to incrementally build a GradeBook class that represents an instructor's grade book and performs various calculations based on a set of stu-

dent grades, such as calculating the average grade, finding the maximum and minimum, and printing a bar chart.

- *Unified Modeling Language™ 2 (UML 2)*. The Unified Modeling Language (UML) has become the preferred graphical modeling language for designers of object-oriented systems. We use UML class diagrams to visually represent classes and their inheritance relationships, and we use UML activity diagrams to demonstrate the flow of control in each of C++’s control statements. We use six types of UML diagrams in the optional OOD/UML ATM case study
- *Compilation and Linking Process for Multiple-Source-File Programs*. Chapter 3 includes a detailed diagram and discussion of the compilation and linking process that produces an executable program.
- *Function Call Stack Explanation*. In Chapter 6, we provide a detailed discussion (with illustrations) of the function call stack and activation records to explain how C++ is able to keep track of which function is currently executing, how automatic variables of functions are maintained in memory and how a function knows where to return after it completes execution.
- *Tuned Treatment of Inheritance and Polymorphism*. Chapters 12–13 have been carefully tuned using an Employee class hierarchy to make the treatment of inheritance and polymorphism clear and accessible for students who are new to OOP.
- *Discussion and Illustration of How Polymorphism Works “Under the Hood.”* Chapter 13 contains a detailed diagram and explanation of how C++ can implement polymorphism, virtual functions and dynamic binding internally. This gives students a solid understanding of how these capabilities really work.
- *Standard Template Library (STL)*. This might be one of the most important topics in the book in terms of your appreciation of software reuse. The STL defines powerful, template-based, reusable components that implement many common data structures and algorithms used to process those data structures. Chapter 22 introduces the STL and discusses its three key components—containers, iterators and algorithms. We show that using STL components provides tremendous expressive power, often reducing many lines of code to a single statement.
- *ISO/IEC C++ Standard Compliance*. We’ve audited our presentation against the most recent ISO/IEC C++ standard document.
- *Debugger Appendices*. We provide two Using the Debugger appendices on the book’s Companion Website—Appendix H, Using the Visual Studio Debugger, and Appendix I, Using the GNU C++ Debugger.
- *Code Testing on Multiple Platforms*. We tested the code examples on various popular C++ platforms including GNU C++ on Linux and Microsoft and Visual C++ on Windows. For the most part, the book’s examples port to popular standard-compliant compilers.

We believe that this book and its support materials will give you an informative, interesting, challenging and entertaining C++ educational experience.

As you read the book, if you have questions, send an e-mail to deitel@deitel.com; we’ll respond promptly. For updates on this book and the status of all supporting C++ software, and for the latest news on all Deitel publications and services, visit www.deitel.com.

Optional Case Study: Using the UML 2 to Develop an Object-Oriented ATM Design

The optional Software Engineering Case Study in Chapters 25 and 26 presents a carefully paced introduction to object-oriented design using the UML. It will help you prepare for the kinds of substantial projects you'll encounter in industry. We design and fully implement the software for a simple automated teller machine (ATM). The case study has been reviewed through many editions by a distinguished team of OOD/UML academics and industry professionals, including leaders in the field from Rational (the creators of the UML) and the Object Management Group (responsible for evolving the UML).

We introduce a simple, concise subset of the UML, then guide you through a first design experience intended for the novice. The case study is not an exercise—it's an end-to-end learning experience that concludes with a detailed walkthrough of the complete C++ code (850 lines).

At the end of Chapter 1, we introduce basic concepts and terminology of OOD. In Chapter 25, we analyze a typical requirements document that specifies a system to be built, determine the objects needed to implement that system, determine the attributes these objects need to have, determine the behaviors these objects need to exhibit, and specify how the objects must interact with one another to meet the system requirements. In Chapter 26, we include a complete C++ code implementation of the ATM, using key object-oriented programming notions, including classes, objects, encapsulation, visibility, composition, inheritance and polymorphism.

Companion Website

We include a set of free, web-based student supplements to the book—the Companion Website—available with new books purchased from Pearson (see the scratch card at the front of the book for your access code). To access the Companion Website, visit www.pearsonhighered.com/deitel/ and select the Companion Website link in the section for this book. *If the access code in front of your book is already redeemed, you can purchase access to this material directly from the Companion Website.*

The Companion Website contains the following chapters and appendices in searchable PDF format:

- Chapter 23, Boost Libraries, Technical Report 1 and C++0x
- Chapter 24, Other Topics
- Chapter 25, ATM Case Study, Part 1: Object-Oriented Design with the UML
- Chapter 26, ATM Case Study, Part 2: Implementing an Object-Oriented Design
- Chapter 27, Game Programming with Ogre
- Appendix F, C Legacy Code Topics
- Appendix G, UML 2: Additional Diagram Types
- Appendix H, Using the Visual Studio Debugger
- Appendix I, Using the GNU C++ Debugger

The Companion Website also includes:

- VideoNotes in which you can watch and listen as Paul Deitel shows you the important features of the code examples in Chapters 2–13 and portions of Chapters 16 and 17.

- Two true/false questions per section with answers for self-review.
- Solutions to approximately half of the solved exercises in the book.

The following additional materials are posted at both the Companion Website and at www.deitel.com/books/cpphttp7/:

- An arrays of pointers to functions example and additional function pointer exercises (from Chapter 8).
- String Class Operator Overloading Case Study (from Chapter 11).
- Building Your Own Compiler exercise descriptions (from Chapter 20).

Dependency Chart

The chart on the next page shows the dependencies among the chapters to help instructors plan their syllabi. *C++ How to Program, 7/e* is appropriate for CS1 and CS2 courses.

Teaching Approach

C++ How to Program, 7/e, contains a rich collection of examples. The book concentrates on the principles of good software engineering and stresses program clarity. We teach by example. We are educators who teach leading-edge programming languages and software-related topics in academic, government, military and industry classrooms worldwide.

Live-Code Approach. *C++ How to Program, 7/e*, is loaded with “live-code” examples. Most new concepts are presented in the context of complete working C++ applications, followed by one or more executions showing program inputs and outputs.

Syntax Coloring. For readability, we syntax color all the C++ code, similar to the way most C++ integrated-development environments and code editors syntax color code. Our syntax-coloring conventions are as follows:

```

comments appear in green
keywords appear in dark blue
errors appear in red
constants and literal values appear in light blue
all other code appears in black

```

Code Highlighting. We place yellow rectangles around key code segments.

Using Fonts for Emphasis. We place the key terms and the index’s page reference for each defining occurrence in **bold maroon** 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;`).

Web Access. All of the source-code examples are available for download from:

```

www.deitel.com/books/cpphttp7/
www.pearsonhighered.com/deitel/

```

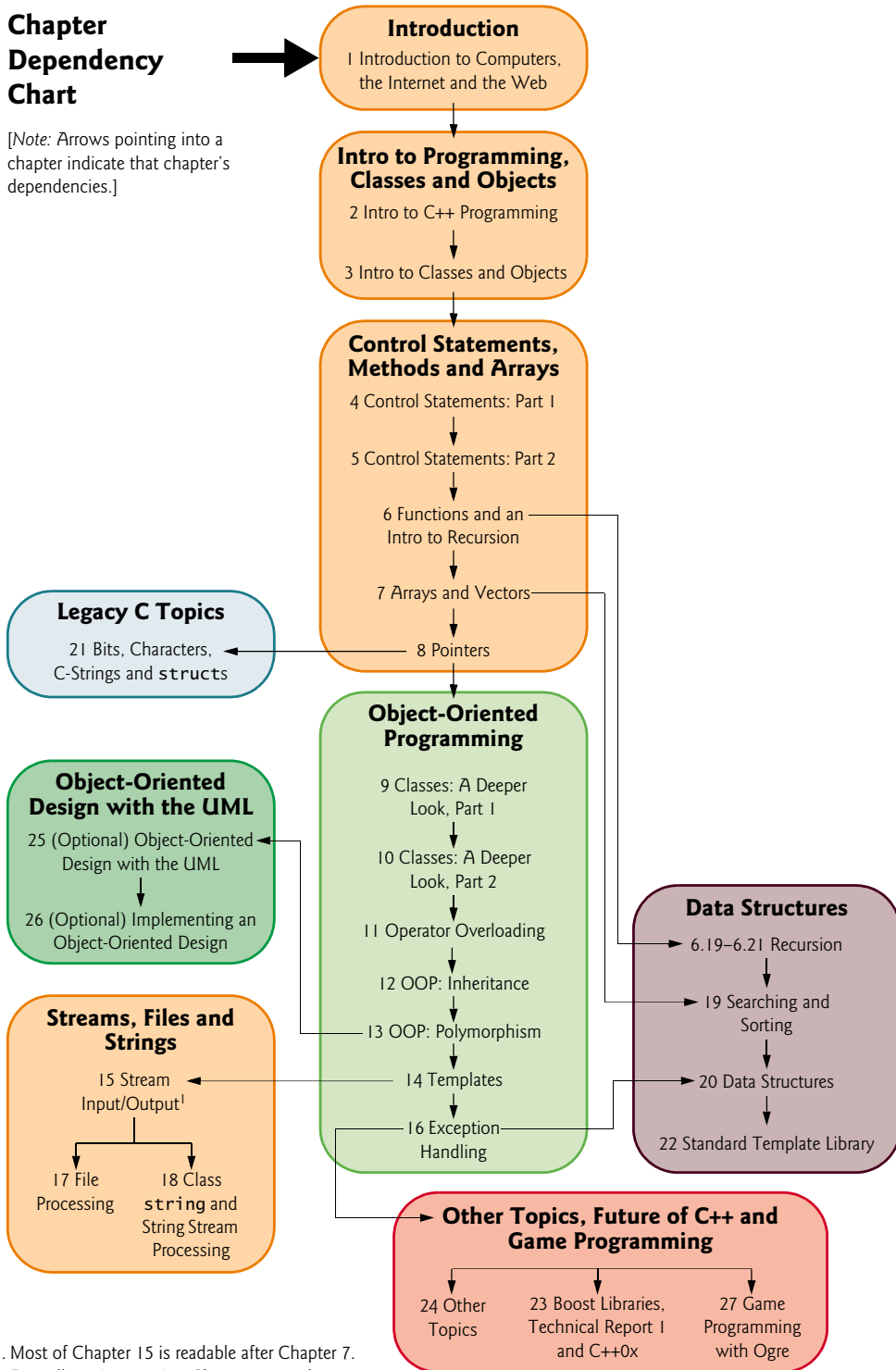
Quotations. Each chapter begins with quotations. We hope that you enjoy relating these to the chapter material.

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

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

Chapter Dependency Chart

[Note: Arrows pointing into a chapter indicate that chapter's dependencies.]



1. Most of Chapter 15 is readable after Chapter 7. A small portion requires Chapters 12 and 14.

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.

Wrap-Up Section. Each chapter ends with a recap of the chapter content, then transitions to the next chapter.

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. All of the exercises in the optional ATM case study are fully solved.

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 C++ statements,
- writing small portions of functions and classes,
- writing complete C++ functions, classes and programs, and
- major projects.

Instructors can use these exercises to form homework assignments, short quizzes, major examinations and term projects. [NOTE: Please do not write to us requesting access to the Pearson Instructor's Resource Center which contains the book's instructor supple-

ments, including the exercise solutions. Access is limited strictly to college instructors teaching from the book. Instructors may obtain access only through their Pearson representatives. Solutions are *not* provided for “project” exercises.] Check out our Programming Projects Resource Center (www.deitel.com/ProgrammingProjects/) for lots of additional exercise and project possibilities.

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 maroon** page number.

Student Resources

Many C++ development tools are available. We wrote *C++ How to Program, 7/e* primarily using Microsoft’s free Visual C++ Express Edition (which is available free for download at www.microsoft.com/express/vc/) and the free GNU C++ (gcc.gnu.org/install/binaries.html), which is already installed on most Linux systems and can be installed on Mac OS X and Windows systems as well. You can learn more about Visual C++ Express at msdn.microsoft.com/vstudio/express/visualc. You can learn more about GNU C++ at gcc.gnu.org. Apple includes GNU C++ in their Xcode development tools, which Mac OS X users can download from developer.apple.com/tools/xcode.

For additional resources and software downloads see our C++ Resource Center:

www.deitel.com/cplusplus/

For other C++ compilers that are available free for download:

www.thefreecountry.com/developercity/ccompilers.shtml
www.compilers.net/Dir/Compilers/CCpp.htm

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 is also environmentally sound and offers students a high-quality digital version of the text for as much as 50% off the cost of a print copy of the text. 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 Supplements

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

- *Solutions Manual* with solutions to most of the end-of-chapter exercises and Lab Manual exercises. We’ve also graded the difficulty of each programming exercise in the book, and have posted a file that assigns a grade level to each programming exercise (in increasingly challenging order: 1, 2, 3 and Project).
- *Test Item File* of multiple-choice questions (approximately two per book section)
- Customizable PowerPoint® slides containing all the code and figures in the text, plus bulleted items that summarize the key points in the text

If you are not already a registered faculty member, contact your Pearson representative or visit www.pearsonhighered.com/educator/replocator/.

Deitel® Buzz Online Free E-mail Newsletter

The *Deitel® Buzz Online* e-mail newsletter will keep you posted about issues related to *C++ How to Program, 7/e*. It also includes commentary on industry trends and developments, links to free articles and resources from our published books and upcoming publications, product-release schedules, errata, challenges, anecdotes, information on our corporate instructor-led training courses and more. To subscribe, visit

www.deitel.com/newsletter/subscribe.html

The Deitel Online Resource Centers

Our website www.deitel.com provides more than 100 Resource Centers on various topics including programming languages, software development, Web 2.0, Internet business and open-source projects—see the list of Resource Centers in the first few pages of this book and visit www.deitel.com/ResourceCenters.html. We've found many exceptional resources online, including tutorials, documentation, software downloads, articles, blogs, podcasts, videos, code samples, books, e-books and more—most of them are free. Each week we announce our latest Resource Centers in our newsletter, the *Deitel® Buzz Online*. Some of the Resource Centers you might find helpful while studying this book are C++, C++ Boost Libraries, C++ Game Programming, Visual C++, UML, Code Search Engines and Code Sites, Game Programming and Programming Projects.

Follow Deitel on Twitter and Facebook

To receive updates on Deitel publications, Resource Centers, training courses, partner offers and more, follow us on Twitter®

[@deitel](https://twitter.com/deitel)

and join the Deitel & Associates group on Facebook®

<http://www.deitel.com/deitelfan/>

Acknowledgments

It's a pleasure to acknowledge the efforts of people whose names do not appear on the cover, but whose hard work, cooperation, friendship and understanding were crucial to the book's production. Many people at Deitel & Associates, Inc., devoted long hours to this project—thanks especially to Abbey Deitel and Barbara Deitel.

We would also like to thank the participants of our Honors Internship program who contributed to this publication—Matthew Pearson, a Computer Science graduate of Cornell University; and Christine Chen, an Operations Research and Information Engineering major at Cornell University.

We are fortunate to have worked on this project with the dedicated team of publishing professionals at Pearson. We appreciate the efforts of Marcia Horton, Editorial Director of Pearson's Engineering and Computer Science Division, and Michael Hirsch, Editor-in-Chief of Computer Science. Carole Snyder recruited the book's review team and managed the review process. Francesco Santalucia (an independent artist) and Kristine Carney of

Pearson designed the book's cover—we provided the concept, and they made it happen. Scott Disanno and Bob Engelhardt managed the book's production. Erin Davis and Margaret Waples marketed the book through academic and professional channels.

C++ How to Program, 7/e Reviewers

We wish to acknowledge the efforts of our reviewers. Adhering to a tight time schedule, they scrutinized the text and the programs and provided countless suggestions for improving the accuracy and completeness of the presentation:

Academic Reviewers:

- Thomas J. Borrelli (Rochester Institute of Technology)
- Peter J. DePasquale (The College of New Jersey)
- Jack Hagemester (Washington State University)
- Williams M. Higdon (University of Indiana)
- Dean Mathias (Utah State University)
- Robert A. McLain (Tidewater Community College)
- Dave Topham (Ohlone College)

Industry Reviewers:

- Chris Cox (Adobe Systems)
- Gregory Dai (Kernel Development)
- Doug Gregor (Apple, Inc.)
- April Reagan (Microsoft)
- José Antonio González Seco (Parliament of Andalusia, Spain)

Well, there you have it! Welcome to the exciting world of C++ and object-oriented programming. We hope you enjoy this look at contemporary computer programming. 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/cpphttp7/`

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

Paul Deitel
Harvey Deitel
Maynard, Massachusetts
July 2009

About the Authors

Paul J. Deitel, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of MIT's Sloan School of Management, where he studied Information Technology.

Through Deitel & Associates, Inc., he has delivered C++, C, Java, C#, Visual Basic and Internet programming courses to industry clients, including Cisco, IBM, 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, Open Environment Corporation, One Wave, Hyperion Software, Adra Systems, Entergy, Cable-Data Systems, Nortel Networks, Puma, iRobot, Invensys and many more. He holds the Java Certified Programmer and Java Certified Developer certifications and has been designated by Sun Microsystems as a Java Champion. He has also lectured on Java and C++ for the Boston Chapter of the Association for Computing Machinery. He and his co-author, Dr. Harvey M. Deitel, are the world's best-selling programming-language textbook authors.

Dr. Harvey M. Deitel, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 48 years of academic and industry 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., with his son, Paul J. Deitel. He and Paul are the co-authors of dozens of books and multimedia packages and they are writing many more. With translations published in Japanese, German, Russian, Traditional Chinese, Simplified Chinese, Spanish, Korean, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish, the Deitels' texts have earned international recognition. Dr. Deitel has delivered hundreds of professional seminars to major corporations, academic institutions, government organizations and the military.

About Deitel & Associates, Inc.

Deitel & Associates, Inc., is an internationally recognized authoring and corporate training organization specializing in computer programming languages, Internet and web software technology, object-technology education and Internet business development. The company provides instructor-led courses delivered at client sites worldwide on major programming languages and platforms, such as C++, Visual C++[®], C, Java[™], Visual C#[®], Visual Basic[®], XML[®], Python[®], object technology, Internet and web programming, and a growing list of additional programming and software-development-related courses. The founders of Deitel & Associates, Inc., are Paul J. Deitel and Dr. Harvey M. Deitel. The company's clients include many of the world's largest companies, government agencies, branches of the military, and academic institutions. Through its 33-year publishing partnership with Prentice Hall/Pearson, Deitel & Associates, Inc., publishes leading-edge programming textbooks, professional books, interactive multimedia *Cyber Classrooms*, *LiveLessons* DVD-based and web-based video courses, and e-content for popular course-management systems. Deitel & Associates, Inc., and the authors can be reached via e-mail at:

deitel@deitel.com

To learn more about Deitel & Associates, Inc., its publications and its *Dive Into*[®] Series Corporate Training curriculum delivered at client locations worldwide, visit:

www.deitel.com/training/

and subscribe to the free *Deitel*[®] *Buzz Online* e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

Individuals wishing to purchase Deitel books, and *LiveLessons* DVD and web-based training courses 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.prenhall.com/mishtml/support.html#order.

