Live in fragments no longer, only connect.
—Edgar Morgan Foster

Welcome to Java and Java How to Program, Eighth Edition! This book presents leading-edge computing technologies for students, instructors, software developers and IT professionals.

We use the Deitel signature “live-code approach,” presenting most concepts in the context of complete working Java programs, rather than using 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/jhtp8/.

At Deitel & Associates, we write programming-language textbooks and professional books for Pearson/Prentice Hall, deliver corporate training courses worldwide and develop Web 2.0 Internet businesses. We have updated the previous edition of this book based on recent changes to the Java language and the evolving preferred ways of teaching and learning programming. All of the chapters have been significantly tuned.

New and Updated Features

Here are the updates we’ve made for Java How to Program, 8/e:

- The book has a new interior design that graphically organizes, clarifies and highlights the information and enhances the book’s pedagogy.

- We updated the entire book to Java Standard Edition 6 Update 11 and carefully audited the manuscript against the Java Language Specification.

- We added the “Making a Difference” exercises set: Students want to make a difference. We’re encouraging them to associate computers and the Internet with solving problems that really matter to individuals, communities, countries and the world. We hope that our new exercises encourage students to think for themselves as they explore complex social issues. These exercises are not intended to make a political statement. They are meant to increase awareness of important issues the world is facing. Students should approach these issues in the context of their own values, politics and beliefs. Many of the new exercises require students to do research on the web—and weave the results into their problem-solving process. Here’s a list of the 34 new “Making a Difference” exercises:

  - Test Drive: Carbon Footprint Calculator
  - Test Drive: Body Mass Index Calculator
  - Attributes of Hybrid Vehicles
  - Gender Neutrality
  - Body Mass Index Calculator
  - World Population Growth Calculator
  - Car Pool Savings Calculator
  - Target Heart Rate Calculator
  - Computerization of Health Records
  - Enforcing Privacy with Cryptography
Preface

We tuned the optional Object-Oriented Design/UML 2 automated teller machine (ATM) case study and reorganized it into two optional chapters (12 and 13) that present the ATM’s design and complete code implementation. The ATM is a nice business example that all students can relate to. In our experience, teaching these two chapters as a unit helps students tie together many of the object-oriented concepts they learn in Chapters 1–10. A key concept in object-oriented programming is the interactions among objects. In most programming textbooks, the code examples create and use one or two objects. The ATM gives students the opportunity to study interactions of many objects that provide the functionality of a substantial system. Chapters 12 and 13 provide complete solutions to all of their exercises. Previously, the case study was distributed through Chapters 2–8, 10 and an appendix. For instructors who wish to cover the case study in a distributed manner, for each section in Chapters 12 and 13, we indicate after which early chapter that section can be covered.

We reinforced our early classes and objects pedagogy, paying careful attention to the guidance of the college instructors on our review teams to ensure that we got the conceptual level right. The treatment of OOP is clear and accessible. We introduce the basic concepts and terminology of object technology in Chapter 1. Students develop their first customized classes and objects in Chapter 3. Presenting objects and classes early gets students “thinking about objects” immediately and mastering these concepts more thoroughly.

We reordered our presentation of data structures. We now begin with generic class ArrayList in Chapter 7. Because students will understand basic generics concepts so early in the book, our later data structures discussions provide a deeper treatment of generic collections—showing how to use the built-in collections of the Java API. We then show how to implement generic methods and classes. Finally, we show how to build custom generic data structures.

We added coverage of Java Web Start and the Java Network Launch Protocol (JNLP), which enable both applets and applications to be launched via a web browser. In addition, the user can install them as shortcuts on the desktop to execute them in the future without revisiting the website. Programs can also request the user’s permission to access local system resources such as files—enabling you
to develop more robust applets and applications that execute safely using Java’s sandbox security model, which applies to downloaded code.

- We reordered several chapters to facilitate teaching the book in modules. The dependencies chart (page xxx) was updated to reflect the new modularization.
- We’ve added many links to online documentation where students can learn more about a class or topic. And we’ve added many links to the Deitel Java-related Resource Centers available at www.deitel.com/ResourceCenters.html.
- Chapter 7 now covers class Arrays—which contains methods for performing common array manipulations—and class ArrayList—which implements a dynamically resizable array-like data structure. This follows our philosophy of using existing classes before learning how to define your own classes.
- We now introduce class BigInteger for arbitrarily large integer values in Chapter 18, Recursion.
- We carefully tuned all the chapters with a focus on increasing clarity and simplicity, eliminating redundancy, reducing page count (this new edition is 90 in-book pages shorter than the previous one), and improving pedagogy and modular organization.
- We’ve replaced all uses of StringTokenizer with the recommended String method split throughout the book. Class StringTokenizer is still discussed, primarily for backward compatibility with legacy code.
- We include an alphabetized list of the important terms defined in each chapter with the page number of the term’s defining occurrence. Defining occurrences are also highlighted in the index with a bold, maroon page number.

All of this has been carefully reviewed by 24 distinguished academics and industry developers who worked with us on Java How to Program, 8/e.

We believe that this book and its support materials will give students and professionals an informative, interesting, challenging and entertaining Java educational experience. We provide a suite of ancillary materials that will help instructors maximize their students’ learning 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 Java software, and for the latest news on all Deitel publications and services, visit www.deitel.com. Sign up at www.deitel.com/newsletter/subscribe.html for the free Deitel® Buzz Online e-mail newsletter, and check out our growing list of Java and related Resource Centers at www.deitel.com/ResourceCenters.html. Each week we announce our latest Resource Centers in the newsletter.

Other Features
Other features of Java How to Program, 8/e, include:

- We audited the presentation against the ACM/IEEE curriculum recommendations and the Computer Science Advanced Placement Examination.
- The early classes and objects presentation features Time, Employee and GradeBook class case studies that weave their way through multiple sections and chapters, gradually introducing deeper OO concepts.
Preface

Instructors teaching introductory courses have a broad choice of the amount of GUI and graphics to cover—from none, to a ten-brief-sections introductory sequence, to a deep treatment in Chapters 14, 15 and 25, and Appendix I.

Our object-oriented programming and design presentations use the UML™ (Unified Modeling Language™)—the industry-standard graphical language for modeling object-oriented systems.

We provide several substantial object-oriented web programming case studies.

Chapter 28, Accessing Databases with JDBC, covers JDBC 4 and uses the Java DB/Apache Derby and MySQL database management systems. The chapter features an OO case study on developing a database-driven address book that demonstrates prepared statements and JDBC 4’s automatic driver discovery.

Chapter 29, JavaServer™ Faces Web Applications, and Chapter 30, Ajax-Enabled JavaServer™ Faces Web Applications, introduce JavaServer Faces (JSF) technology and use it with Netbeans 6.5 to build web applications quickly and easily. Chapter 29 includes examples on building web application GUIs, handling events, validating forms and session tracking. Chapter 30 discusses developing Ajax-enabled web applications, using JavaServer Faces technology. The chapter features a database-driven multistier web address book application that allows users to add and search for contacts. This Ajax-enabled application gives the reader a nice sense of Web 2.0 software development. The application uses Ajax-enabled JSF components to suggest contact names while the user types a name to locate.

Chapter 31, Web Services, uses a tools-based approach to creating and consuming SOAP- and REST-based web services. Case studies include developing blackjack and airline reservation web services.

We use a new tools-based approach for rapidly developing web applications; all the tools are available free for download.

We provide 100+ Resource Centers (www.deitel.com/resourcecenters.html) to support our academic and professional readers. Their topics include Java SE 6, Java, Java Assessment and Certification, Java Design Patterns, Java EE 5, Code Search Engines and Code Sites, Game Programming, Programming Projects and many more. Sign up at www.deitel.com/newsletter/subscribe.html for the free Deitel® Buzz Online e-mail newsletter—each week we announce our latest Resource Center(s) and include other items of interest to our readers.

We discuss key software engineering community concepts, such as Web 2.0, Ajax, SaaS (Software as a Service), web services, open-source software, design patterns, mashups, refactoring, agile software development, rapid prototyping and more.

We completely reworked Chapter 26, Multithreading [special thanks to Brian Goetz and Joseph Bowbeer—co-authors of Java Concurrency in Practice, Addison-Wesley, 2006].

We discuss the SwingWorker class for developing multithreaded user interfaces.

We discuss the GroupLayout layout manager in the context of the GUI design tool in the NetBeans IDE.
We present \texttt{JTable} sorting and filtering capabilities which allow the user to re-sort the data in a \texttt{JTable} and filter it by regular expressions.

We discuss the \texttt{StringBuilder} class, which performs better than \texttt{StringBuffer} in non-threaded applications.

We present annotations, which greatly reduce the amount of code you have to write to build applications.

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

UML 2 has become the preferred graphical modeling language for designing object-oriented systems. We use UML activity diagrams (in preference to flowcharts) to demonstrate the flow of control in each of Java’s control statements, and we use UML class diagrams to visually represent classes and their inheritance relationships.

We include an optional (but highly recommended) case study on object-oriented design using the UML. 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). In the case study, we design and fully implement the software for a simple automated teller machine (ATM). The optional Software Engineering Case Study in Chapters 12 and 13 presents a carefully paced introduction to object-oriented design using the UML.

We introduce a simple, concise subset of the UML 2, then guide the reader through a first design experience intended for the novice. The case study is not an exercise; rather, it’s an end-to-end learning experience that concludes with a detailed walkthrough of the complete Java code. Chapters 12 and 13 help students develop an object-oriented design to complement the object-oriented programming concepts they’ve learned in Chapters 1 through 11.

At the end of Chapter 1, we introduce basic concepts and terminology of OOD. In Chapter 12, we consider more substantial issues, as we undertake a challenging problem with the techniques of OOD. 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 13, we include a complete Java code implementation of the object-oriented system that we designed in Chapter 12.

This case study helps prepare students for the kinds of substantial projects they’ll encounter in industry. We employ a carefully developed, incremental object-oriented design process to produce a UML 2 model for our ATM system. From this design, we produce a substantial working Java implementation using key object-oriented programming notions, including classes, objects, encapsulation, visibility, composition, inheritance and polymorphism.

**Dependency Chart**

The chart on the next page shows the dependencies among the chapters to help instructors plan their syllabi. *Java How to Program, 8/e*, is appropriate for a variety of programming courses at various levels, most notably CS 1 and CS 2 courses and introductory course
Preface

Chapter Dependency Chart

[Note: Arrows pointing into a chapter indicate that chapter’s dependencies. Some chapters have multiple dependencies.]

Introduction
1 Introduction to Computers, the Internet and the Web

Intro to Programming, Classes and Objects
2 Intro to Java Applications
3 Intro to Classes and Objects

Control Statements, Methods and Arrays
4 Control Statements: Part 1
5 Control Statements: Part 2
6 Methods: A Deeper Look
7 Arrays and ArrayLists

Object-Oriented Programming
8 Classes and Objects: A Deeper Look
9 Object-Oriented Programming: Inheritance
10 Object-Oriented Programming: Polymorphism
11 Exception Handling

Object-Oriented Design with the UML
12 (Optional) Object-Oriented Design with the UML
13 (Optional) Implementing an Object-Oriented Design

Strings and Files
16 Strings, Characters and Regular Expressions
17 Files, Streams and Object Serialization

Multithreading and Networking
26 Multithreading
27 Networking

Database-Driven Desktop and Web Development
28 JDBC
29 JSF Web Applications
30 Ajax-Enabled JSF Web Applications
31 Web Services

GUI & Graphics Track
3.9 Using Dialog Boxes
4.14 Creating Simple Drawings
5.10 Drawing Rectangles and Ovals
6.13 Colors and Filled Shapes
7.15 Drawing Arcs
8.16 Using Objects with Graphics
9.8 Displaying Text and Images Using Labels
10.8 Drawing with Polymorphism

GUI, Graphics, Applets and Multimedia
14 GUI Components: Part 1
15 Graphics and Java2D
23 Applets and Java Web Start
24 Multimedia: Applets and Applications
25 GUI Components: Part 2

Data Structures
18 Recursion
19 Searching, Sorting and Big O
20 Generic Collections
21 Generics
22 Custom Generic Data Structures

Multithreading and Networking
26 Multithreading
27 Networking

1. Chapter 18 is dependent on Chapters 14 and 15 for GUI and graphics used in one example.
2. Chapter 26 is dependent on Chapter 14 for GUI used in one example and on Chapters 20–21 for one example.
3. Chapter 27 is dependent on Chapter 23 for one example that uses an applet. The large case study at the end of this chapter depends on Chapter 25 for GUI and Chapter 26 for multithreading.
4. Chapter 28 is dependent on Chapter 14 for GUI used in one example.
sequences in related disciplines. The book has a clearly delineated, modular organization. Chapters 1–11 and 14–17 form an accessible elementary programming sequence with a solid introduction to object-oriented programming. Optional Chapters 12–13 form an accessible introduction to object-oriented design with the UML. The GUI and Graphics Track and Chapters 14, 15, 23, 24 and 25 form a substantial GUI, graphics and multimedia sequence. Chapters 18–22 form a nice data-structures sequence. Chapters 26–27 form a solid introduction to multithreading and Internet networking. Chapters 28–31 form a clear database-intensive web development sequence.

**Syllabus Assistance**
We’re happy to assist instructors designing syllabi based on *Java How to Program, 8/e*. You can reach us by email (deitel@deitel.com) or phone (1 978 823-0130). We’ll respond promptly.

**Teaching Approach**
*Java How to Program, 8/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 government, industry, military and academic classrooms worldwide.

**Live-Code Approach.** *Java How to Program, 8/e*, is loaded with “live-code” examples. By this we mean that each new concept is presented in the context of a complete working Java application, followed immediately by one or more actual executions showing the program’s inputs and outputs.

**Syntax Coloring.** For readability, we syntax color all the Java code, similar to the way most Java 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 easier reference. We emphasize on-screen components in the **bold Helvetica** font (e.g., the **File** menu) and emphasize Java program text in the **Lucida** font (for example, `int x = 5;`).

**Web Access.** All of the source-code examples for *Java How to Program, 8/e*, are available for download from:

- [www.deitel.com/books/jhtp8](http://www.deitel.com/books/jhtp8)

**Objectives.** Each chapter begins with a statement of objectives.

**Quotations.** The learning objectives are followed by quotations. We hope that you enjoy relating these to the chapter material.
Illustrations/ Figures. Abundant charts, tables, line drawings, programs and program output are included. We model the flow of control in control statements with UML activity diagrams. UML class diagrams model the fields, constructors and methods of classes. We make extensive use of six major UML diagram types in the optional OOD/UML 2 ATM case study.

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.

Wrap-Up Section. Each chapter ends with a “wrap-up” section that recaps the chapter content and transitions to the next chapter.

Summary Bullets. Each chapter ends with additional pedagogical devices. We present a section-by-section, bullet-list-style summary of the chapter.

Terminology. We include an alphabetized list of the important terms defined in each chapter with the page number of the term’s defining occurrence. Defining occurrences are also highlighted in the index with a bold, maroon page number.
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 program statements; writing small portions of methods and Java classes; writing complete methods, Java classes and programs; and building major term 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. Access is limited strictly to college instructors teaching from the book. Instructors may obtain access only through their Pearson representatives.] Be sure to check out our Programming Projects Resource Center (http://www.deitel.com/ProgrammingProjects/) for lots of additional exercise and project possibilities.

Thousands of Index Entries. We have included an extensive index, which is especially useful when you use the book as a reference.

Student Resources Included with Java How to Program, 8/e

Many Java development tools are available for purchase, but you need none of these to get started with Java. For Windows systems, all the software you’ll need for this book is available free for download from the web or on the accompanying CD. For other platforms, all the software you’ll need for this book is generally available free for download from the web. We wrote most of the examples in Java How to Program, 8/e, using the free Java Standard Edition Development Kit (JDK) 6. The current JDK version (and separately its documentation) can be downloaded from Sun’s Java website java.sun.com/javase/downloads/index.jsp. Mac OS X users can download Java from developer.apple.com/java. In several chapters, we also used the Netbeans IDE. Netbeans is available as a bundle with the JDK from the preceding Sun Java website, or you can download it separately from www.netbeans.org/downloads/index.html. The Eclipse IDE can be downloaded from www.eclipse.org/downloads/ . MySQL can be downloaded from dev.mysql.com/downloads/connector/j/5.1.html . You can find additional resources and software downloads in our Java SE 6 Resource Center at:

www.deitel.com/JavaSE6Mustang/

The CD that accompanies Java How to Program, 8/e, contains versions of the following software packages for use on Microsoft® Windows®:

- Java™ SE Development Kit (JDK) 6 Update 11—which was used to create and test all the programs in the book.
- Eclipse IDE for Java EE Developers 3.4.1.
- NetBeans™ IDE Version 6.5 All Bundle.
- MySQL® 5.0 Community Server version 5.0.67.
- MySQL® Connector/J version 5.1.7.

Netbeans and Eclipse are integrated development environments (IDEs) for developing all types of Java applications. MySQL and MySQL Connector/J are provided for the database.
applications in Chapters 28–31. All of these tools are downloadable for other platforms also, as we discuss in Before You Begin, after this Preface.

The CD also contains a web page with links to the Deitel & Associates, Inc., website and the Pearson website. This web page can be loaded into a web browser to afford quick access to all the resources.

**Java Multimedia Cyber Classroom, 8/e**

*Java How to Program, 8/e*, includes a free, web-based, video-intensive interactive multimedia ancillary to the book—the *Java Multimedia Cyber Classroom, 8/e*—available with new books purchased from Pearson. Our web-based *Cyber Classroom* includes video walkthroughs of all the code examples in Chapters 1–11 and some of the code examples in Chapters 14 and 17, solutions to about half of the exercises in the book, a lab manual and more. For more information about the web-based *Cyber Classroom*, please visit

www.prenhall.com/deitel/cyberclassroom/

Students like the *Cyber Classroom*’s interactivity and reference capabilities. Professors tell us that their students enjoy using the *Cyber Classroom* and consequently spend more time on the courses, mastering more of the material than in textbook-only courses. Instructors may want to explicitly weave *Cyber Classroom* assignments into their syllabi.

**Instructor Resources for Java How to Program, 8/e**

The Pearson Instructor’s Resource Center contains

- the *Solutions Manual* with solutions to most of the end-of-chapter exercises,
- a *Test Item File* of multiple-choice questions (approximately two per book section) and
- PowerPoint® slides containing all the code and figures in the text, plus bulleted items that summarize the key points in the text. Instructors can customize the slides.

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

**Computer Science AP Courses**

*Java How to Program, 8/e*, is a suitable textbook for teaching AP Computer Science classes and for preparing students to take the AP exam.

**Deitel® Buzz Online Free E-mail Newsletter**

Each week, the *Deitel® Buzz Online* announces our latest Resource Center(s) and 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. It’s also a good way to keep posted about issues related to *Java How to Program, 8/e*. 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. The Resource Centers evolve out of the research we do to support our books and business endeavors. 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 (www.deitel.com/newsletter/subscribe.html). Some of the Resource Centers you might find helpful while studying this book are Java SE 6, Java, Java Assessment and Certification, Java Design Patterns, Java EE 5, Code Search Engines and Code Sites, Game Programming, Programming Projects and many more.

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 new edition—Nicholas Doiron, an Electrical and Computer Engineering major at Carnegie Mellon University; and Matthew Pearson, a Computer Science major at Cornell University.

We are fortunate to have worked on this project with the talented and dedicated team of publishing professionals at Pearson. We appreciate the extraordinary efforts of Marcia Horton, Editorial Director of Pearson’s Engineering and Computer Science Division. Carole Snyder and Dolores Mars did an extraordinary job recruiting the book’s review team and managing the review process. Francesco Santalucia (an independent artist) and Kristine Carney of Pearson did a wonderful job designing the book’s cover—we provided the concept, and they made it happen. Scott Disanno and Robert Engelhardt did a marvelous job managing the book’s production. Our marketing manager Erin Davis and her boss Margaret Waples did a great job marketing the book through academic and professional channels.

Java How to Program, 8/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:

Sun Microsystems Reviewers:
• Lance Andersen
• Soundararajan Angusamy
• Lawrence Prem Kumar
• Simon Ritter
• Sang Shin
• Alexander Zuev
Preface

Academic Reviewers:
- William E. Duncan (Louisiana State University)
- Diana Franklin (University of California, Santa Barbara)
- Edward F. Gehringer (North Carolina State University)
- Ric Heishman (George Mason University)
- Patty Kraft (San Diego State University)
- Manjeet Rege, Ph.D. (Rochester Institute of Technology)
- Tim Margush (University of Akron)
- Sue McFarland Metzger (Villanova University)
- Shyamal Mitra (The University of Texas at Austin)
- Susan Rodger (Duke University)
- Amr Sabry (Indiana University)
- Monica Sweat (Georgia Tech)

Industry Reviewers:
- Joseph Bowbeer (Consultant)
- Peter Pilgrim (Lloyds TSB)
- José Antonio González Seco (Parliament of Andalusia)
- S. Sivakumar (Astra Infotech Private Limited)
- Raghavan “Rags” Srinivas (Intuit)
- Vinod Varma (Astra Infotech Private Limited)

Well, there you have it! Java is a powerful programming language that will help you write programs quickly and effectively. It 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/jHTP8/

We hope you enjoy reading *Java How to Program, 8/e*, as much as we enjoyed writing it!

*Paul J. Deitel  
Dr. Harvey M. Deitel*

**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. He holds the Java Certified Programmer and Java Certified Developer certifications and has been designated by Sun Microsystems as a Java Champion. Through Deitel & Associates, Inc., he has delivered Java, C, C++, 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, Stratus, Cambridge Technol-
yogy Partners, Open Environment Corporation, One Wave, Hyperion Software, Adra Sys-
tems, Entergy, CableData Systems, Nortel Networks, Puma, iRobot, Invensys and many
more. 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 20
years of college teaching experience, including earning tenure and serving as the Chairman
of the Computer Science Department at Boston College before founding Deitel & Asso-
ciates, 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
Traditional Chinese, Simplified Chinese, Japanese, German, Russian, Spanish, Korean,
French, Polish, Italian, Portuguese, Greek, Urdu and Turkish, the Deitels’ texts have
earned international recognition. Dr. Deitel has delivered hundreds of professional semi-
nars to major corporations, academic institutions, government organizations and the mil-
itary.

About Deitel & Associates, Inc.

Deitel & Associates, Inc., is an internationally recognized corporate training and author-
ing organization specializing in computer programming languages, Internet and web soft-
ware technology, object-technology education and Internet business development
through its Web 2.0 Internet Business Initiative. The company provides instructor-led
courses on major programming languages and platforms, such as Java™, C++, C, Visual
C#®, Visual Basic®, Visual C++, 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. Har-
vey M. Deitel. The company’s clients include many of the world’s largest companies, gov-
ernment 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 pop-
ular 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 worldwide 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 gov-

Preface

ernment, the military and academic institutions should be placed directly with Pearson. For more information, visit www.prenhall.com/mischtm/support.html#order.

Check out the growing list of online Deitel Resource Centers at:

www.deitel.com/resourcecenters.html

Individuals wishing to purchase Deitel publications can do so through:

www.deitel.com/books/index.html

Bulk orders by corporations, the government, the military and academic institutions should be placed directly with Pearson. For more information, visit

www.prenhall.com/mischtm/support.html#order