Preface

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

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

The new Chapter 1 engages students with intriguing facts and figures to get them excited about studying computers and programming. The chapter includes a table of some of the research made possible by computers; current technology trends and hardware discussion; the data hierarchy; a table of mobile and Internet app platforms; a new section on social networking; an introduction to Android; a table of popular web services; 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.

The book is appropriate for introductory course sequences based on the ACM/IEEE curriculum recommendations and for AP Computer Science exam preparation.

We focus on software engineering best practices. At the heart of the book is the Deitel signature "live-code approach"—concepts are presented in the context of complete working programs, rather than in code snippets. Each complete code example is accompanied by live sample executions. All the source code is available at `www.deitel.com/books/jhtp9/` and at the book's Companion Website `www.pearsonhighered.com/deitel/`.

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, visit `www.deitel.com/books/jhtp9/`, follow us on Facebook (`www.deitel.com/deitelfan`) and Twitter (`@deitel`), and subscribe to the *Deitel® Buzz Online* newsletter (`www.deitel.com/newsletter/subscribe.html`).

New and Updated Features

Here are the updates we've made for *Java How to Program, 9/e*:

Java Standard Edition (SE) 7

- ***Easy to use as a Java SE 6 or Java SE 7 book.*** There are a few Java Standard Edition (SE) 7 features that affect CS 1 and CS 2 courses. We cover those features in optional modular sections that are easy to include or omit. Here's some of the new functionality: `Strings` in `switch` statements, the `try`-with-resources statement for managing `AutoClosable` objects, multi-`catch` for defining a single exception handler to replace multiple exception handlers that perform the same task, the NIO filesystem APIs and inferring the types of generic objects from the variable they're assigned to by using the <> notation. We also overview the new concurrency API features.

- ***Java SE 7 filesystem APIs.*** We provide an *alternate* online version of Chapter 17, Files, Streams and Object Serialization, that's reimplemented with the new filesystem APIs from Java SE 7.
- ***Java SE 7's `AutoClosable` versions of `Connection`, `Statement` and `ResultSet`.*** With the source code for Chapter 28, Accessing Databases with JDBC, we provide a version of the chapter's first example that's implemented using Java SE 7's `AutoClosable` versions of `Connection`, `Statement` and `ResultSet`. `AutoClosable` objects reduce the likelihood of resource leaks when you use them with Java SE 7's `try`-with-resources statement, which automatically closes the `AutoClosable` objects allocated in the parentheses following the `try` keyword.

Pedagogic Features

- ***Enhanced Making a Difference exercises set.*** We encourage you to use computers and the Internet to research and solve significant social problems. These exercises are meant to increase awareness and discussion of important issues the world is facing. We hope you'll approach them with your own values, politics and beliefs. Check out our new Making a Difference Resource Center at `www.deitel.com/MakingADifference` for additional ideas you may want to investigate further.
- ***Page numbers for key terms in chapter summaries.*** For key terms that appear in the chapter summaries, we include the page number of the key term's defining occurrence.
- ***VideoNotes.*** The Companion Website includes extensive VideoNotes in which co-author Paul Deitel explains in detail most of the programs in the core chapters. Instructors have told us that their students find the VideoNotes valuable.

Object Technology

- ***Object-oriented programming and design.*** 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. [For courses that require a late-objects approach, consider *Java How to Program, Late Objects Version, 8/e,* which begins with six chapters on programming fundamentals (including two on control statements) and continues with seven chapters that gradually introduce object-oriented programming concepts.]
- ***Exception handling.*** We integrate basic exception handling earlier in the book and instructors can easily pull more material forward from Chapter 11, Exception Handling: A Deeper Look.
- ***Class `Arrays` and `ArrayList`.*** Chapter 7 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 getting lots of practice *using existing classes while learning how to define your own classes.*
- ***OO case studies.*** 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.

- ***Optional Case Study: Using the UML to Develop an Object-Oriented Design and Java Implementation of an ATM.*** The UML™ (Unified Modeling Language™) is the industry-standard graphical language for modeling object-oriented systems. Chapters 12–13 include an *optional* case study on object-oriented design using the UML. We design and implement the software for a simple automated teller machine (ATM). We analyze a typical requirements document that specifies the system to be built. We determine the classes needed to implement that system, the attributes the classes need to have, the behaviors the classes need to exhibit and specify how the classes must interact with one another to meet the system requirements. From the design we produce a *complete* Java implementation. Students often report having a "light-bulb moment"—the case study helps them "tie it all together" and really understand object orientation.
- ***Reordered data structures presentation.*** We begin with generic class `ArrayList` in Chapter 7. Because *students will understand basic generics concepts 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.

Database and Web Development

- ***JDBC 4.*** 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.
- ***Java Server Faces (JSF) 2.0.*** Chapters 29–30 have been updated to introduce JavaServer Faces (JSF) 2.0 technology, which greatly simplifies building JSF web applications. Chapter 29 includes examples on building web application GUIs, validating forms and session tracking. Chapter 30 discusses data-driven and Ajax-enabled JSF applications. The chapter features a database-driven multitier web address book 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.
- ***Web services.*** Chapter 31, Web Services, demonstrates creating and consuming SOAP- *and* REST-based web services. Case studies include developing blackjack and airline reservation web services.
- ***Java Web Start and the Java Network Launch Protocol (JNLP).*** We introduce Java Web Start and JNLP, which enable applets *and* applications to be launched via a web browser. Users can install locally for later execution. 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.

Multithreading

- ***Multithreading.*** We completely reworked Chapter 26, Multithreading [special thanks to the guidance of Brian Goetz and Joseph Bowbeer—two of the co-authors of *Java Concurrency in Practice,* Addison-Wesley, 2006].
- ***`SwingWorker` class.*** We use class `SwingWorker` to create *multithreaded user interfaces*.

GUI and Graphics

- ***Scalable GUI and graphics presentation.*** Instructors teaching introductory courses have a broad choice of the amount of GUI and graphics to cover—from none, to an optional 10-brief-sections introductory sequence woven in with the early chapters, to a deep treatment in Chapters 14, 15 and 25, and Appendix I.
- ***`GroupLayout` layout manager.*** We discuss the `GroupLayout` layout manager in the context of the GUI design tool in the NetBeans IDE.
- ***`JTable` sorting and filtering capabilities.*** Chapter 28 uses these capabilities to re-sort the data in a `JTable` and filter it by regular expressions.

Other Features

- ***Android.*** Because of the tremendous interest in Android-based smartphones and tablets, we've included a three-chapter introduction to Android app development on the Companion Website. These chapters are from our new Deitel Developer Series book *Android for Programmers: An App-Driven Approach*. After you learn Java, you'll find it straightforward to develop and run Android apps on the free Android emulator that you can download from `developer.android.com`.
- ***Software engineering community concepts***. We discuss agile software development, refactoring, design patterns, LAMP, SaaS (Software as a Service), PaaS (Platform as a Service), cloud computing, open-source software and more.

Dependency Chart

The chart on the next page shows the dependencies among the chapters to help instructors plan their syllabi. *Java How to Program, 9/e*, is appropriate for a variety of programming courses at various levels, most notably CS 1 and CS 2 courses and introductory course 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 rich database-intensive web application development sequence.

Teaching Approach

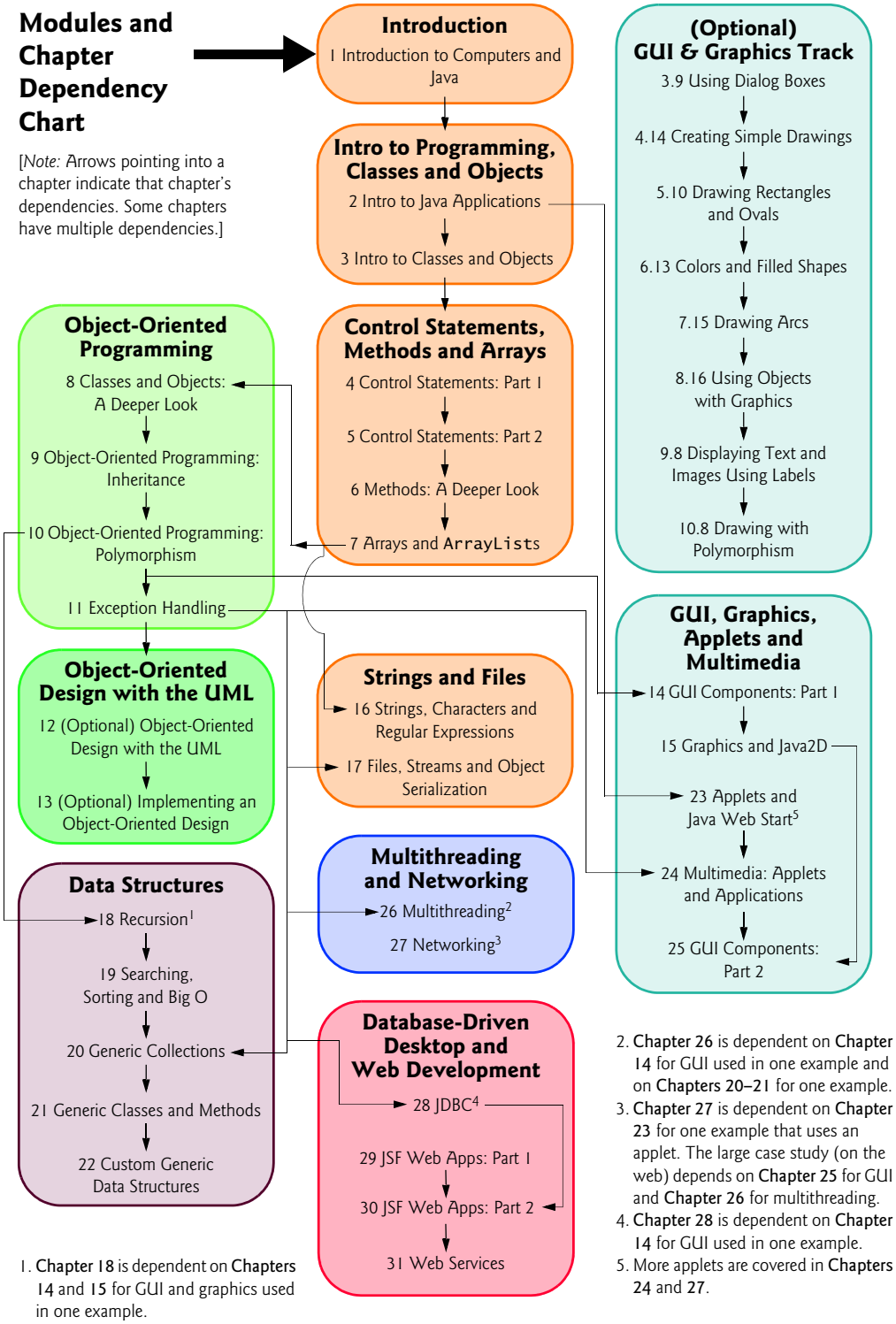
Java How to Program, 9/e, contains hundreds of complete working examples. We stress program clarity and concentrate on building well-engineered software.

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
```

Modules and Chapter Dependency Chart

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



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 (on the web) 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.
5. More applets are covered in **Chapters 24** and **27**.

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 can be downloaded from:

```
www.deitel.com/books/jhtp9
www.pearsonhighered.com/deitel
```

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

Illustrations/Figures. Abundant tables, line drawings, UML diagrams, programs and program outputs 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 bugs and removing them from your programs; many describe aspects of Java 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.*

Look-and-Feel Observations

The Look-and-Feel Observations *highlight graphical-user-interface conventions. These observations help you design attractive, user-friendly graphical user interfaces that conform to industry norms.*

Summary Bullets. We present a section-by-section bullet-list summary of the chapter. For ease of reference, we include the page number of each key term's defining occurrence in the text.

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. The chapter exercises include:

- simple recall of important terminology and concepts
- What's wrong with this code?
- What does this code do?
- writing individual statements and small portions of methods and classes
- writing complete methods, classes and programs
- major projects
- in many chapters, Making a Difference exercises.

Index. We've included an extensive index. Defining occurrences of key terms are highlighted with a **bold maroon** page number.

Software Used in *Java How to Program, 9/e*

All the software you'll need for this book is available free for download from the web. See the Before You Begin section that follows this Preface for links to each download.

We wrote most of the examples in *Java How to Program, 9/e*, using the free Java Standard Edition Development Kit (JDK) 6. For the optional Java SE 7 modules, we used the OpenJDK's early access version of JDK 7. In Chapters 29–31, we also used the Netbeans IDE, and in Chapter 28, we used MySQL and MySQL Connector/J. You can find additional resources and software downloads in our Java Resource Centers at:

```
www.deitel.com/ResourceCenters.html
```

Java IDE Resource Kit

Your instructor may have ordered through your college bookstore a Value Pack edition of *Java How to Program, 9/e* that comes bundled with the Java IDE Resource Kit. This kit contains CD or DVD versions of Java™ SE Development Kit 6 for Windows®, Eclipse™ IDE for Windows®, NetBeans™ IDE, jGRASP™ IDE, DrJava IDE, BlueJ IDE and the TextPad® Text Editor for Windows®. Free versions of these IDEs also can be downloaded from the web. The Java IDE Resource Kit also includes access to a Companion Website containing step-by-step VideoNotes and written instructions to help you get started with each development environment. If your book did not come with the Java IDE Resource Kit, you can purchase access to the Resource Kit's Companion Website from `www.pearsonhighered.com/javaidekit/`. You'll still need to download the free software separately.

Discounts on *Deitel Developer Series* Books

If you'd like to receive information on professional *Deitel Developer Series* titles, including *Android for Programmers: An App-Driven Approach*, please register your copy of *Java How*

to Program, 9/e at `informit.com/register`. You'll receive information on how to purchase *Android for Programmers* at a discount.

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 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`):

- ***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 the vast majority of the end-of-chapter exercises.

Please do not write to us requesting access to the Pearson Instructor's Resource Center which contains the book's instructor supplements, 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 for lots of additional exercise and project possibilities (`www.deitel.com/ProgrammingProjects/`).

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

Acknowledgments

We'd like to thank Abbey Deitel and Barbara Deitel for long hours devoted to this project. We're fortunate to have worked on this project 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 recruited the book's reviewers and managed the review process. Bob Engelhardt managed the book's production.

Reviewers

We wish to acknowledge the efforts of our eighth and ninth edition reviewers. They scrutinized the text and the programs and provided countless suggestions for improving the presentation: Lance Andersen (Oracle), Soundararajan Angusamy (Sun Microsystems), Joseph Bowbeer (Consultant), William E. Duncan (Louisiana State University), Diana Franklin (University of California, Santa Barbara), Edward F. Gehringer (North Carolina State University), Huiwei Guan (Northshore Community College), Ric Heishman (George Mason University), Dr. Heinz Kabutz (JavaSpecialists.eu), Patty Kraft (San Diego State University), Lawrence Premkumar (Sun Microsystems), Tim Margush (Univer-

sity of Akron), Sue McFarland Metzger (Villanova University), Shyamal Mitra (The University of Texas at Austin), Peter Pilgrim (Consultant), Manjeet Rege, Ph.D. (Rochester Institute of Technology), Manfred Riem (Java Champion, Consultant, Robert Half), Simon Ritter (Oracle), Susan Rodger (Duke University), Amr Sabry (Indiana University), José Antonio González Seco (Parliament of Andalusia), Sang Shin (Sun Microsystems), S. Sivakumar (Astra Infotech Private Limited), Raghavan "Rags" Srinivas (Intuit), Monica Sweat (Georgia Tech), Vinod Varma (Astra Infotech Private Limited) and Alexander Zuev (Sun Microsystems).

Well, there you have it! As you read the book, we'd appreciate your comments, criticisms, corrections and suggestions for improvement. Please address all correspondence to:

```
deitel@deitel.com
```

We'll respond promptly. We hope you enjoy working with *Java How to Program, 9/e*. Good luck!

Paul and Harvey Deitel

About the Authors

Paul J. 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 Java, C++, C, C#, Visual Basic and Internet 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 authors.

Dr. Harvey M. 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., with his son, Paul J. Deitel. He and Paul are the co-authors of dozens of books and LiveLessons multimedia packages and they are writing many more. The Deitels' texts have earned international recognition, with translations published in Japanese, German, Russian, Chinese, Spanish, Korean, 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., is an internationally recognized corporate training and authoring organization. The company provides instructor-led courses delivered at client sites worldwide on major programming languages and platforms, such as Java™, C++, Visual C++®, C, Visual C#®, Visual Basic®, XML®, Python®, object technology, Internet and web programming, Android™ and iPhone® app development, and a growing list of additional programming and software-development 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 35-year publishing partnership with Prentice Hall/Pearson, Deitel & Associates, Inc., publishes leading-edge programming textbooks, professional books and *LiveLessons* DVD-based and web-based video courses. 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
```

and follow the authors on Facebook (`www.deitel.com/deitelfan`) and Twitter (`@deitel`).

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.pearsonhighered.com
```