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

Welcome to the Java programming language and Java How to Program, Late Objects Version, Eighth Edition! This book is appropriate for introductory-level programming courses (commonly known as CS1 and CS2). It also provides substantial depth for those who already know how to program and want to learn Java.

At the heart of this book is the Deitel signature “live-code approach.” Concepts are presented 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/jhtp8LOV/.

Motivation for Java How to Program, Late Objects Version, 8/e

There are several approaches for teaching first courses in Java programming. The two most popular are the late objects approach and the early objects approach. To meet these diverse needs, we have published two versions of this book:

• Java How to Program, Late Objects Version, 8/e, and
• Java How to Program, 8/e (which offers an early objects approach).

The key difference between these two editions is the order in which topics are presented in Chapters 1–7. The books have virtually identical content from Chapters 8 to 31. In Java How to Program, Late Objects Version, 8/e, Chapters 26–31 are available only in PDF format at the book’s companion website.

Chapters 1–6 in Java How to Program, Late Objects Version, 8/e, form the core of a pure-procedural programming CS1 course that covers operators, data types, input/output, control statements, methods, arrays, strings and files. Instructors who want to introduce some object-oriented programming in a first course can include a portion of the next several chapters.

New Features

If you’ve seen the seventh edition of Java How to Program, you might wonder how this book differs. Besides the organizational changes previously mentioned, here are updates we’ve made for Java How to Program, Late Objects Version, 8/e:

• We updated the book to the latest version of Java Standard Edition 6.
• We added the Making a Difference exercises set to help you discover how you can make a difference in the world in which you live. We encourage you to associate computers and the Internet with solving problems that really matter to individuals, communities, countries and the world. These new exercises will encourage...
Preface

you to think critically as you explore complex social issues. Approach these issues in the context of your own values, politics and beliefs. Many of the new exercises require you to do research on the web and weave the results into your problem-solving process. There are more than 30 Making a Difference exercises throughout the book on subjects including global warming, population growth, gender neutrality, computerization of health records, alternative tax plans, hybrid vehicles, accessibility for people with disabilities and more. A complete list of Making a Difference exercises is available at

www.deitel.com/books/JHTP8LOV/

• We refreshed the book’s interior design in a manner that graphically organizes, clarifies and highlights the information and enhances the book’s pedagogy.

• We reinforced our object-oriented programming 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—this section can be deferred to the beginning of Chapter 7 by instructors who wish to teach pure procedural programming in the early chapters. You’ll develop your first customized classes and objects in Chapter 7.

• We reordered several chapters to facilitate teaching the book in modules. The chapter dependency chart (page xxxii) shows the new modularization.

• We’ve added many links to the online Java documentation and the Deitel Java-related Resource Centers available at www.deitel.com/ResourceCenters.html.

• Chapter 6 now covers class Arrays—which contains methods for performing common array manipulations, such as searching and sorting an array’s contents—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 introduce string and file processing at the end of Chapter 6. This enables instructors to teach these topics early in a first course if they wish. We cover these topics in more depth in Chapters 16 and 17, respectively.

• We’ve replaced all uses of StringTokenizer with the preferred String method split, which uses Java’s robust regular-expression processing capabilities. Class StringTokenizer is still discussed, primarily for backward compatibility with legacy code.

• We reordered our presentation of data structures. We now begin with generic class ArrayList in Chapter 6 (this section can be deferred to Chapter 7) enabling students to understand basic collections and 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 custom generic methods and classes. Finally, we show how to build custom generic data structures.

• 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 students 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 7–10. 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.

- 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 now introduce class BigInteger for arbitrarily large integer values in Chapter 18.
- The chapter terminology sections now include the page number of each key term’s defining occurrence.

Other Features

Other features of Java How to Program, Late Objects Version, 8/e, include:

- We audited the presentation against the ACM/IEEE curriculum recommendations and the Computer Science Advanced Placement Examination.
- The classes and objects presentation features class Time, class Employee and other class case studies, some of which weave their way through multiple sections and chapters, gradually introducing deeper OO concepts.
- Instructors teaching introductory courses have a broad choice of the amount of GUI and graphics to cover—from none, to an optional eight-brief-sections introductory sequence spread over the early chapters, to a deep treatment in Chapters 14, 15, 23, 24 and 25, and Appendix J. (See the section Optional GUI and Graphics Track for more information.)
- 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, mostly for use in second programming courses and by professionals.
- Chapter 28 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.
- Chapters 29 and 30 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 multitier 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 implement “type ahead” capability that suggests contact names as the user types a name to locate.

- Chapter 31 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 and more.
- We reworked Chapter 26 [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 JTable sorting and filtering capabilities which allow the user to sort the data in a JTable and filter it by regular expressions.
- We discuss the StringBuilder class, which performs better than StringBuffer in non-threaded applications.
- We present annotations, which greatly reduce the amount of code you have to write to build applications.

Optional GUI and Graphics Track

In Java How to Program, Late Objects Version, 8/e, students learn how to create their own classes starting in Chapter 7. The optional sections of the GUI and Graphics Track give students the opportunity to create objects of existing classes and do interesting things with them before the students attempt to create their own customized classes. The optional sections are:

- Section 2.9: Using Dialog Boxes
- Section 3.14: Creating Simple Drawings

Optional Case Study: Designing an Object-Oriented ATM

- Section 4.10: Drawing Rectangles and Ovals
- Section 5.13: Colors and Filled Shapes
- Section 6.20: Drawing Arcs
- Section 8.16: Using Objects with Graphics
- Section 9.8: Displaying Text and Images Using Labels
- Section 10.8: Drawing with Polymorphism

Instructors who are teaching pure procedural programming in the first course can skip them. Instructors teaching the first course with some object-oriented programming have several options. They can defer the GUI and Graphics Track sections from Chapters 2–6 until after Chapter 7. Or they can avoid these sections altogether and simply cover the treatment of GUI and graphics that begins in Chapter 14.

Optional Case Study: Designing an Object-Oriented ATM

The UML 2 has become the preferred graphical modeling language for designing object-oriented systems. We use industry standard UML activity diagrams (in preference to the older 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.

The optional Software Engineering Case Study in Chapters 12 and 13 presents a carefully paced introduction to object-oriented design using the UML—we design and fully implement the software for a simple automated teller machine (ATM). The case study has been reviewed by academics and industry professionals, including leaders in the field from Rational (the creators of the UML; Rational is now part of IBM) and the Object Management Group (an industry group now responsible for evolving the UML).

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

We introduce a concise subset of the UML 2, then guide you through the design process. The case study is not an exercise—it’s an end-to-end learning experience that concludes with a detailed walkthrough of the complete Java code.

At the end of Chapter 1, we introduce some basic concepts and terminology of OOD. In Chapter 12, we consider more substantial design 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.

Dependency Chart

The chart on the next page shows the dependencies among the chapters to help instructors plan their syllabi. *Java How to Program, Late Objects Version, 8/e* is appropriate for a variety of programming courses at various levels, most notably CS1 and CS2 courses and
Preface

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 at the end of Chapter 27 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.
introductory course sequences in related disciplines. The book has a convenient modular organization.

**Introduction to Computers and Programming**
Chapters 1–2 discuss computers, the Internet, the web, then introduce the basics of Java programming, including input/output, arithmetic operators and decision making.

**Control Statements, Methods and Arrays**
Chapters 3–6 form a clear introduction to procedural programming in Java, covering data types, operators, control statements, methods, arrays, and introducing strings and files. This is the non-object-oriented core of the book’s late objects approach.

**Object-Oriented Programming**
Chapters 7–11 provide a solid introduction to object-oriented programming, covering classes, objects, inheritance, polymorphism, interfaces and exceptions.

**(Optional) Object-Oriented Design with the UML**
Optional Chapters 12–13 form an accessible introduction to object-oriented design with the UML.

**GUI, Graphics, Applets and Multimedia**
The GUI and Graphics Track (optional sections in Chapters 2–6 and 8–10) and Chapters 14, 15, 23, 24 and 25 form a substantial GUI, graphics and multimedia sequence.

**Strings and Files**
Chapters 16–17 continue our coverage of string and file processing that begins in Chapter 6.

**Data Structures**
Chapters 18–22 form a nice data-structures sequence, including recursion, searching, sorting, using generic collections, defining generic methods and classes, and building custom generic data structures.

**Multithreading and Networking**
Chapters 26–27 form a solid introduction to multithreading and Internet networking.

**Database-Driven Desktop and Web Development**
Chapters 28–31 form a database-intensive web development sequence, including coverage of JDBC, JavaServer Faces (JSF), Ajax and web services.

**Teaching Approach**
*Java How to Program, Late Objects Version, 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 academic, government, military and industry classrooms worldwide.

**Live-Code Approach.** *Java How to Program, Late Objects Version, 8/e,* is loaded with “live-code” examples. By this we mean that most new concepts are presented in the context of
Preface

complete working Java applications, followed immediately by one or more actual executions showing program inputs and outputs.

Syntax Shading. For readability, we syntax shade all the Java code, similar to the way most Java integrated-development environments and code editors syntax color code. Our syntax-shading conventions are as follows:

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

Code Highlighting. We place gray 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 blue 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 (e.g., int x = 5;).

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

www.deitel.com/books/jhtp8LOV

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, 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 use six major UML diagram types in the optional OOD/UML 2 ATM case study in Chapters 12 and 13.

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 bugs and removing them from your programs; many describe aspects of Java that prevent bugs from getting into programs in the first place.

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.

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 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 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.] Be sure to check out our Programming Projects Resource Center (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. Defining occurrences of key terms are highlighted with a bold, blue page number.

Student Resources

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

Preface

web. We wrote most of the examples 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/, and MySQL Connector/J 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 the book 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.

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). The Companion Website includes video walkthroughs (called Video Notes) of all the code examples in Chapters 2–11 and some of the code examples in Chapters 14 and 17, solutions to about half of the exercises in the book, and a lab manual.

For more information about the Companion Website, please visit

www.pearsonhighered.com/deitel/

Instructors may want to explicitly weave Video Notes assignments into their syllabi.

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
- **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/relocator/.

Computer Science AP Courses

Java How to Program, Late Objects Version, 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

The Deitel® Buzz Online e-mail newsletter will keep you posted about issues related to Java How to Program, Late Objects Version, 8/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. The Resource Centers evolve out of the research we do to support our publications 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. 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 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

and join the Deitel & Associates group on Facebook®

www.facebook.com/group.php?gid=38765799996

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, and Michael Hirsch, Editor-in-Chief of Computer Science. 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.

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:

- William E. Duncan (Louisiana State University)
- Diana Franklin (University of California, Santa Barbara)
- Edward F. Gehring (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)

Sun Microsystems Reviewers:

- Lance Andersen
- Soundararajan Angusamy
- Lawrence Prem Kumar
- Simon Ritter
About the Authors

- Sang Shin
- Alexander Zuev

Other 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’d 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/jhtp8LOV/

We hope you enjoy reading *Java How to Program, Late Objects Version, 8*th edition, 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, SunGard Higher Education, Stratus, Cambridge Technology Partners, Open Environment Corporation, One Wave, Hyperion Software, Adra Systems, 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 Department of Computer Science. His research interests include computer architecture, computer security, computer networking, computer systems, database systems, expert systems, intelligent agents, operating systems, programming languages, and software engineering.
Preface

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 Traditional Chinese, Simplified Chinese, Japanese, German, Russian, Spanish, Korean, French, Polish, 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 corporate training and authoring 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 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. 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/mischtm/support.html#order.

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

www.deitel.com/resourcecenters.html