



CD includes (for Windows/Linux):
Java™ 2 Platform Development Kit
Standard Edition 5.0, BlueJ™ 1.3.5,
JCreator LE 3.10 (Windows Only), IDE 3.0,
Edit 4.1, IGRASP™ 1.7.0, NetBeans™ IDE 3.0,
MySQL™ 4.0.20, MySQL™ Connector/J 3.0.14,
Apache Tomcat 5.0.25

Java大学教程

——(第六版)(英文版)——

Java How to Program, Sixth Edition



Featuring **EARLY CLASSES + OBJECTS**
8:00 AM OOD with the UML™ 2
9:00 AM J2SE™ 5.0



EARLY CLASSES/OBJECTS/OOP

- INHERITANCE, POLYMORPHISM
- INTERFACES, NESTED CLASSES
- OPTIONAL GUI, GRAPHICS TRACK
- OOP CASE STUDIES

J2SE™ 5.0 COMPLIANCE

- SCANNER, PRINTF, ENHANCED FOR
- ENUM, BOXING, VARARGS
- ROWSETS, STATIC IMPORT
- CONCURRENCY APIS

FUNDAMENTALS

- I/O, APPLICATIONS, APPLET
- TYPES, ARRAYS, OPERATORS
- METHODS, CONTROL STATEMENTS
- USING API DOCS, DEBUGGER

JAVA APIS

- SWING GUI, EVENT HANDLING
- EXCEPTIONS, STRINGS, REGEX
- GRAPHICS, JAVA 2D™, THREADS
- FILES, STREAMS, MULTIMEDIA

DATA STRUCTURES

- RECURSION, SEARCHING, SORTING
- DATA STRUCTURES, COLLECTIONS
- GENERICS (J2SE™ 5.0)

INTERNET/WEB APPLICATIONS

- NETWORKING
- JDBC™, SERVLETS, JSP™
- DEITEMESSENGER CHAT APP

OOD/UML™ 2 ATM CASE STUDY

- DIAGRAMS: CLASS, ACTIVITY, USE CASE, COMMUNICATION, SEQUENCE, STATE

电子工业出版社
PUBLISHING HOUSE OF ELECTRONIC INDUSTRY
<http://www.phei.com.cn>

DEITEL®

[美] H. M. Deitel 著
P. J. Deitel

Java 大学教程 (第六版) (英文版)

Java How to Program, Sixth Edition

The complete, authoritative DEITEL LIVE-CODE introduction to object-oriented programming with the new Java™ 2 Platform Standard Edition 5.0, JDBC™, Servlets and JSP™!

Java™ is the most popular object-oriented programming language with four million developers! This new edition of the world's most widely used Java textbook uses an **early classes and objects approach**.

"Does an excellent job describing the new features included in JDK 5.0..." —Doug Kohlert, Sun Microsystems

"The UML material is very well done and a huge asset." —Gavin Osborne, Saskatchewan Institute

"Introduces OOP early without burying the reader in complexity. The conceptual level is perfect. This will be a great help the next time I teach 101. ... I was introduced to JHTP by my students who pleaded with me to drop our assigned text in favor of JHTP. No other text comes close to its quality of organization and presentation. The Live-Code approach to presenting exemplary code makes a big difference in the learning outcome."

—Walt Bunch, Chapman University

"I teach Java programming and object-oriented analysis and design. The optional UML/OOD case study is the best presentation of the ATM example I have seen. ... Classes are introduced early. The authors separate the functionality and the testing of the class, emphasizing good program design principles."

—Craig W. Slinkman, U. of Texas-Arlington

"A well-written presentation of object-based programming principles. A must-read for every aspiring programmer."

—Marita Ellixson, Egin AFB, Univ. of Arkansas, Indiana Wesleyan Univ., Lead Facilitator

"Of particular note is the authors' attention to good software engineering." —Dean Mellas, Cerritos College

"The optional Graphics & GUI case study track introduces students to some of the more fun things in Java! Working with color, graphics and GUI always gets their attention!"

—Karen Arlien, Bismark State College

"Motivates inheritance well. Progression of examples is very nice." —Stephen Weiss, CS Chair, UNC Chapel Hill

"Probably the most complete coverage of learning through examples in published material today. This material is such high quality—it is unbelievable. The ATM example is super!"

—Anne Horton, AT&T Bell Laboratories

Java How to Program includes comprehensive coverage of object-oriented programming in Java, and several major integrated case studies: the Time class (three sections of Chapter 8), the Employee class (Chapters 9 and 10), the GradeBook class (Chapters 3–5 and 7), the optional OOD/UML ATM system (Chapters 1–8 and 10), the optional GUI and graphics track (Chapters 3–12), the DeitelMessenger instant-messaging application (Chapter 24) and two multitier, database-driven Web applications—Online Survey (Chapter 26) and Guest Book (Chapter 27).

Dr. Harvey M. Deitel and Paul J. Deitel are the founders of **Deitel & Associates, Inc.**, the internationally recognized corporate training and content-creation organization specializing in Java™, C++, C, Visual C#® .NET, Visual Basic® .NET, Visual C++® .NET, XML, Python, Perl, Internet, Web and object technologies. The Deitels are the authors of many international best-selling programming language textbooks, including **Java How to Program, 6/e, Internet & World Wide Web How to Program, 3/e, C How to Program, 4/e** and **C++ How to Program, 4/e**.

Java How to Program's instructor and student resources include Web sites (www.deitel.com and www.prenhall.com/deitel) with the book's code examples (also on the enclosed CD) and information for faculty, students and professionals. Contact the authors at:

www.deitel.com

For information on **DEITEL® DIVE INTO SERIES** on-site corporate training offered worldwide, and to subscribe to the free **DEITEL® BUZZ ONLINE** e-mail newsletter, visit:

www.deitel.com

PRENTICE HALL Upper Saddle River, NJ 07458

www.prenhall.com



ISBN 978-7-121-07021-1



9 787121 070211 >

定价: 118.00 元
(含光盘1张)



责任编辑: 许菊芳

责任美编: 喻晓



本书贴有激光防伪标志, 凡没有防伪标志者, 属盗版图书。

国外计算机科学教材系列

Java 大学教程

(第六版) (英文版)

Java How To Program

Sixth Edition

H. M. Deitel

Deitel & Associates, Inc.

[美]

著

P. J. Deitel

Deitel & Associates, Inc.

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是专为初学 Java 编程的读者准备的。全书共分 29 章及 14 个附录, 前 3 章主要讲述计算机、Internet、Web 及 Java 应用、类及对象等概念, 后面的各章涉及控制语句、方法、数组、继承、多态、GUI 组件、图形与 Java 2D、异常处理、文件与流、递归、查找与排序、数据结构、泛型、集合、applet、多媒体、多线程、网络、JDBC、servlet、JSP、格式化输出、字符串/字符及正则表达式等内容。随书 CD-ROM 中的 9 个附录是对全书的重要补充, 以 7 种不同的小图标列出的提示性文字也是本书的特色之一。

本书适合作为计算机及相关专业的教材, 也适合希望学习 Java 编程的初学者。

Original edition, entitled JAVA HOW TO PROGRAM, SIXTH EDITION, 0131483986 by DEITEL, HARVERY M.; DEITEL, PAUL J., published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2005 by Pearson Education, Inc. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2008.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in People's Republic of China excluding Hong Kong, Macau and Taiwan.

本书由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

此版本仅限在中国大陆 (不包括香港、澳门特别行政区以及台湾地区) 发行与销售。

本书贴有 Pearson Education 培生教育出版集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号: 图字: 01-2008-2046

图书在版编目 (CIP) 数据

Java 大学教程: 第 6 版 = Java How to Program, Sixth Edition: 英文 / (美) 戴特尔著.

北京: 电子工业出版社, 2008.7

(国外计算机科学教材系列)

ISBN 978-7-121-07021-1

I. J... II. 戴... III. JAVA 语言 - 程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 096328 号

责任编辑: 许菊芳

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787 × 1092 1/16 印张: 73.5 字数: 1882 千字

印 次: 2008 年 7 月第 1 次印刷

定 价: 118.00 元

凡所购买电子工业出版社的图书有缺损问题, 请向购买书店调换; 若书店售缺, 请与本社发行部联系。联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

Library of Congress Cataloging-in-Publication Data
On file

Vice President and Editorial Director, ECS: **Marcia J. Horton**
Senior Acquisitions Editor: **Kate Hargett**
Associate Editor: **Jennifer Cappello**
Assistant Editor: **Sarah Parker**
Editorial Assistant: **Michael Giacobbe**
Vice President and Director of Production and Manufacturing, ESM: **David W. Riccardi**
Executive Managing Editor: **Vince O'Brien**
Managing Editor: **Tom Mansbreck**
Production Editor: **John F. Lovell**
Production Editor, Media: **Bob Engelhardt**
Production Assistant: **Asha Robra**
Director of Creative Services: **Paul Belfanti**
A/V Production Editor: **Xiaohong Zhu**
Art Studio: **Artworks, York, PA**
Art Director: **Geoffrey Cassar**
Cover Design: **Harvey M. Deitel, Shawn Murphy, Geoffrey Cassar**
Interior Design: **Harvey M. Deitel, Geoffrey Cassar**
Manufacturing Manager: **Trudy Piscioti**
Manufacturing Buyer: **Lisa McDowell**
Marketing Manager: **Pamela Hersperger**
Marketing Assistant: **Barrie Reinhold**



© 2005 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs and to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or to the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks and registered trademarks. Where those designations appear in this book, and Prentice Hall and the authors were aware of a trademark claim, the designations have been printed in initial caps or all caps. All product names mentioned remain trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-148398-6

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educacion de Mexico, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, Inc., *Upper Saddle River, New Jersey*

Trademarks

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Pearson Education is independent of Sun Microsystems, Inc.

Microsoft, Microsoft® Internet Explorer and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Netscape browser window © 2004 Netscape Communications Corporation. Used with permission. Netscape Communications has not authorized, sponsored, endorsed, or approved this publication and is not responsible for its content.

MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Copyright © 1999–2002 The Apache Software Foundation. All Rights Reserved.

H. M. Deitel

Deitel & Associates, Inc.

P. J. Deitel

Deitel & Associates, Inc.



Upper Saddle River, New Jersey 07458



JAVATM

HOW TO PROGRAM

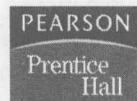
SIXTH EDITION

H. M. Deitel

Deitel & Associates, Inc.

P. J. Deitel

Deitel & Associates, Inc.



Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data
On file

Vice President and Editorial Director, ECS: **Marcia J. Horton**
Senior Acquisitions Editor: **Kate Hargrett**
Associate Editor: **Jennifer Cappello**
Assistant Editor: **Sarah Parker**
Editorial Assistant: **Michael Giacobbe**
Vice President and Director of Production and Manufacturing, ESM: **David W. Riccardi**
Executive Managing Editor: **Vince O'Brien**
Managing Editor: **Tom Mansbreck**
Production Editor: **John F. Lovell**
Production Editor, Media: **Bob Engelhardt**
Production Assistant: **Asha Rohra**
Director of Creative Services: **Paul Belfanti**
A/V Production Editor: **Xiaohong Zhu**
Art Studio: **Artworks, York, PA**
Art Director: **Geoffrey Cassar**
Cover Design: **Harvey M. Deitel, Shawn Murphy, Geoffrey Cassar**
Interior Design: **Harvey M. Deitel, Geoffrey Cassar**
Manufacturing Manager: **Trudy Piscioti**
Manufacturing Buyer: **Lisa McDowell**
Marketing Manager: **Pamela Hersperger**
Marketing Assistant: **Barrie Reinhold**



© 2005 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or to the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks and registered trademarks. Where those designations appear in this book, and Prentice Hall and the authors were aware of a trademark claim, the designations have been printed in initial caps or all caps. All product names mentioned remain trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-148398-6

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educacion de Mexico, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, Inc., *Upper Saddle River, New Jersey*

*In memory of Edsger W. Dijkstra:
We continue to learn from your writings every day.*

Harvey M. Deitel and Paul J. Deitel

Preface

“Live in fragments no longer, only connect.”

— Edgar Morgan Foster

Welcome to Java and *Java How to Program, Sixth Edition!* At Deitel & Associates, we write computer science textbooks and professional books. This book was a joy to create. To start, we put the fifth edition of *Java How to Program* under the microscope:

- We audited the presentation against the most recent ACM/IEEE curriculum recommendations and the Computer Science Advanced Placement Examination.
- All of the chapters have been significantly updated and upgraded.
- We changed to an early classes and objects pedagogy. Now students build their first reusable classes starting in Chapter 3.
- All of the GUI and graphics in the early chapters has been replaced by carefully paced optional sections in Chapters 3–10 with two special exercise sections in Chapters 11 and 12. Instructors have a broad choice of the amount of GUI and graphics to cover—from none, to a 10-section introductory sequence, to a deep treatment in Chapters 11, 12 and 22.
- We updated our object-oriented presentation to use the latest version of the *UML (Unified Modeling Language)*—*UML™ 2.0*—the industry-standard graphical language for modeling object-oriented systems.
- We replaced the optional elevator simulator case study from the previous edition with a new optional OOD/UML automated teller machine (ATM) case study in Chapters 1–8 and 10. The new case study is much simpler and more attractive for first and second programming courses.
- Several multi-section object-oriented programming case studies have been added.
- We incorporated key new features of Sun Microsystems’ latest release of Java—the *Java 2 Platform, Standard Edition version 5.0 (J2SE 5.0)*.
- The design of the book has been completely revised. This new design uses color, fonts and various design elements to enhance a student’s learning experience.

All of this has been carefully scrutinized by a team of 37 distinguished academic and industry reviewers.

We believe that this book and its support materials have everything instructors and students need for an informative, interesting, challenging and entertaining Java educational experience. In this Preface, we overview various conventions used in the book, such as syntax coloring the code examples, “code washing” and code highlighting. We discuss the software bundled with the book as well as the comprehensive suite of educational materials that help instructors maximize their students’ learning experience, including the *Instructor’s Resource CD*, PowerPoint® Slide lecture notes, lab manual, companion Web site, course management systems, SafariX (Pearson Education’s WebBook publications) and more.

As you read this book, if you have questions, send an e-mail to deitel@deitel.com; we will respond promptly. Please visit our Web site, www.deitel.com, regularly and be sure to sign up for the free *Deitel® Buzz Online* e-mail newsletter at www.deitel.com/newsletter/subscribe.html. We use the Web site and the newsletter to keep our readers and industry clients informed of the latest news on Deitel publications and services. Please check the Web site occasionally for errata, updates regarding the Java software, free downloads and other resources.

Features in *Java How to Program, 6/e*

This new edition contains many new and enhanced features including:

Updated for the Java 2 Platform Standard Edition 5.0 (J2SE 5.0)

We updated entire text to reflect the latest release of J2SE 5.0. New topics include:

- obtaining formatted input with class `Scanner`
- displaying formatted output with the `System.out` object's `printf` method
- using enhanced `for` statements to process array elements and collections
- declaring methods with variable-length argument lists (“varargs”)
- using `enum` classes that declare sets of constants
- importing the `static` members of one class for use in another
- converting primitive-type values to type-wrapper objects and vice versa using autoboxing and auto-unboxing, respectively
- using generics to create general models of methods and classes that can be declared once, but used with many different data types
- using the generics-enhanced data structures of the Collections API
- using the Concurrency API to implement multithreaded applications
- using JDBC `RowSets` to access data in a database

In addition, we carefully audited the manuscript against the *Java Language Specification* (available at java.sun.com/docs/books/jls/index.html). The programs you create as you study this text should work with any J2SE 5.0 compatible Java platform.

[Note: Sun Microsystems recently renamed J2SE from the Java 2 Platform, Standard Edition 1.5.0 to the Java 2 Platform, Standard Edition 5.0. However, Sun decided not to replace all occurrences of 1.5.0 with 5.0 in the online Java documentation (available at java.sun.com/j2se/1.5.0/docs/api/index.html) and in the software installation directory (which is called `jdk1.5.0`). Sun's Web site accepts URLs that replace 1.5.0 with 5.0. For example, you can use the URL java.sun.com/j2se/5.0/docs/api/index.html to access the online documentation.]

New Interior Design

Working with the creative services team at Prentice Hall, we redesigned the interior styles for our *How to Program* Series. In response to reader requests, we now place the key terms and the index's page reference for each defining occurrence in **blue, bold style** 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`).

Syntax Coloring

This book is presented in full color to show programs and their outputs as they typically appear on a computer screen^①. We syntax color all the Java code, as most Java integrated-development environments and code editors do. This greatly improves code readability—an especially important goal, given that this book contains 20,383 lines of code. Our syntax-coloring conventions are as follows:

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

Code Highlighting

Extensive code highlighting makes it easy for readers to spot each program's featured segments and helps students review the material rapidly when preparing for exams or labs.

“Code Washing”

“Code washing” is our term for applying comments, using meaningful identifiers, applying uniform indentation conventions and using vertical spacing to highlight significant program units. This process results in programs that are easy to read and self-documenting. We have extensively “code washed” of all the source-code programs in the text and in the book's ancillaries. We have worked hard to make our code exemplary.

① 影印版已取消这些颜色区分。——编者注

Early Classes and Objects Approach

Students are still introduced to the basic concepts and terminology of object technology in Chapter 1. In the previous edition, students began developing their customized classes and objects in Chapter 8, but in this edition, they start doing that in our completely new Chapter 3. Chapters 4–7 have been carefully rewritten from an “early classes and objects approach.” For all intents and purposes, this new edition is object-oriented from the start and throughout the text. Moving the discussion of objects and classes to earlier chapters gets students “thinking about objects” immediately and mastering these concepts more thoroughly. Java is not trivial by any means, but it’s fun to program with, and students can see immediate results. Students can get text-based and graphical programs running quickly by using Java’s extensive class libraries of reusable components.

Carefully Tuned Treatment of Object-Oriented Programming in Chapters 8–10

We performed a high-precision upgrade of *Java How to Program, 5/e*. This edition is clearer and more accessible—especially if you are new to object-oriented programming (OOP). We completely rewrote the OOP chapters with an integrated case study in which we develop an employee payroll hierarchy, and we motivate interfaces with a payables hierarchy.

Case Studies

This book is loaded with case studies that span multiple sections and chapters. Often we build on a class introduced earlier in the book to demonstrate new programming concepts later in the book. Students learn the new concepts in the context of applications that they already know. These case studies include the development of the:

- GradeBook class in Chapters 3, 4, 5 and 7
- ATM application in the optional OOD/UML sections of Chapters 1–8 and 10
- polymorphic drawing program in the optional GUI and Graphics track in Chapters 3–12
- Time class in several sections of Chapter 8
- Employee payroll application in Chapter 9 and Chapter 10

Integrated GradeBook Case Study

To reinforce our early classes presentation, we present an integrated case study using classes and objects in Chapters 3–5 and 7. We incrementally build a `GradeBook` class that represents an instructor’s grade book and performs various calculations based on a set of student grades—finding the average, finding the maximum and minimum, and printing a bar chart. Our goal is to familiarize students with the important concepts of objects and classes through a real-world example of a substantial class. We develop this class from the ground up, constructing methods from control statements and carefully developed algorithms, and adding instance variables and arrays as needed to enhance the functionality of the class.

GUI and Graphics Case Study (Optional)

The optional GUI and Graphics Case Study in Chapters 3–12 demonstrates techniques for adding visual elements to applications. It is designed for those who want to begin learning Java’s powerful capabilities for creating graphical user interfaces (GUIs) and graphics. Each section introduces basic concepts and provides visual, graphical examples and complete source code. In the first few sections, we show how to create simple graphical applications. In the subsequent sections, we use the object-oriented programming concepts presented through Chapter 10 to create an application that draws a variety of shapes polymorphically. In Chapter 11, we add an event-driven GUI interface and in Chapter 12 we incorporate graphics features of Java 2D.

Unified Modeling Language (UML)—Using the UML 2.0 to Develop an Object-Oriented Design of an ATM

The Unified Modeling Language™ (UML) has become the preferred graphical modeling language for designing object-oriented systems. All the UML diagrams in the book comply with the new UML 2.0 specification. We use UML class diagrams to visually represent classes and their inheritance relationships, and we use UML activity diagrams to demonstrate the flow of control in each of Java’s several control statements.

This *Sixth Edition* includes a new, optional (but highly recommended) case study on object-oriented design using the UML. The case study was reviewed by a distinguished team of OOD/UML academic and industry professionals, includ-

ing leaders in the field from Rational (the creators of the UML and now a division of IBM) and the Object Management Group (responsible for maintaining and evolving the UML). In the case study, we design and fully implement the software for a simple automatic teller machine (ATM). The “Software Engineering Case Study” sections at the ends of Chapters 1–8 and 10 present a carefully paced introduction to object-oriented design using the UML. We introduce a concise, simplified subset of the UML 2.0, then guide the reader through a first design experience intended for the novice object-oriented designer/programmer. The case study is not an exercise; rather, it is an end-to-end learning experience that concludes with a detailed walkthrough of the complete Java code. The “Software Engineering Case Study” sections help students develop an object-oriented design to complement the object-oriented programming concepts they begin learning in Chapter 1 and implementing in Chapter 3. In the first of these sections at the end of Chapter 1, we introduce basic concepts and terminology of OOD. In the optional “Software Engineering Case Study” sections at the ends of Chapters 2–5, 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 Appendix J, we include a Java implementation of the object-oriented system that we designed in the earlier chapters. This case study will help prepare students for the kinds of substantial projects they will encounter in industry. We employ a carefully developed, incremental object-oriented design process to produce a UML model for our ATM system. From this design, we produce a substantial working Java implementation using key programming notions, including classes, objects, encapsulation, visibility, composition, inheritance and polymorphism.

Database and Web-Applications Development with JDBC, Servlets and JSP

Chapter 25, Accessing Databases with JDBC, demonstrates how to build data-driven applications with the JDBC™ API. Chapter 26, Servlets, and Chapter 27, JavaServer Pages (JSP), expand our treatment of Internet and Web programming topics, giving readers everything they need to develop their own Web-based applications. Readers will learn how to build so-called *n*-tier applications, in which the functionality provided by each tier can be distributed to separate computers across the Internet or executed on the same computer. In particular, we build a three-tier Web-based survey application and a three-tier Webbased guest-book application. Each application’s information is stored in the application’s data tier—in this book, a database implemented with MySQL (a trial version is on the CD that accompanies this book). The user enters requests and receives responses at each application’s client tier, which is typically a computer running a Web browser such as Microsoft Internet Explorer or Netscape. Web browsers, of course, know how to communicate with Web sites throughout the Internet. The middle tier contains both a Web server and one or more application-specific servlets (Java programs that extend the server to handle requests from client browsers) or JavaServer Pages (an extension of servlets that simplifies handling requests and formatting responses to client browsers). We use Apache’s Tomcat Web server for these examples. Tomcat, which is the reference implementation for the servlets and JavaServer Pages technologies, is included on the CD that accompanies this book and is available free for download from www.apache.org. Tomcat communicates with client Web browsers across the Internet using the HyperText Transfer Protocol (HTTP)—the foundation of the World Wide Web. We discuss the crucial role of the Web server in Web programming and provide extensive examples demonstrating interactions between a Web browser and a Web server.

Teaching Approach

Java How to Program, 6/e contains a rich collection of examples, exercises and projects drawn from many fields to provide the student with a chance to solve interesting real-world problems. The book concentrates on the principles of good software engineering and stresses program clarity. We avoid arcane terminology and syntax specifications in favor of teaching by example. Our code examples have been tested on popular Java platforms. We are educators who teach leading-edge topics in industry classrooms worldwide. Dr. Harvey M. Deitel has 20 years of college teaching experience and 15 years of industry teaching experience. Paul Deitel has 12 years of industry teaching experience and is one of the world’s most experienced Java corporate trainers, having taught about 100 Java courses at all levels since 1996 to government, industry, military and academic clients of Deitel & Associates.

Learning Java via the LIVE-CODE Approach

Java How to Program, 6/e, is loaded with LIVE-CODE examples—each new concept is presented in the context of a complete working Java application that is immediately followed by one or more sample executions showing the program's inputs and outputs. This style exemplifies the way we teach and write about programming. We call this method of teaching and writing the **LIVE-CODE Approach**. *We use programming languages to teach programming languages.* Reading the examples in the text is much like typing and running them on a computer. We provide all the source code for the book's examples on the accompanying CD and at www.deitel.com—making it easy for students to run each example as they study it.

World Wide Web Access

All of the source-code examples for *Java How to Program, 6/e*, (and our other publications) are available on the Internet as downloads from the following Web sites:

www.deitel.com
www.prenhall.com/deitel

Registration is quick and easy, and the downloads are free. We suggest that students download all the examples, then run each program as they read the corresponding text discussions. Making changes to the examples and immediately seeing the effects of those changes is a great way to enhance your Java learning experience.

Objectives

Each chapter begins with a statement of objectives. This lets students know what to expect and gives them an opportunity, after reading the chapter, to determine if they have met these objectives. This is a confidence builder and a source of positive reinforcement.

Quotations

The learning objectives are followed by quotations. Some are humorous, philosophical or offer interesting insights. We hope that you will enjoy relating the quotations to the chapter material. Many of the quotations are worth a second look after reading the chapter.

Outline

The chapter outline helps students approach the material in a top-down fashion, so they can anticipate what is to come, and set a comfortable and effective learning pace.

20,383 Lines of Code in 256 Example Programs (with Program Outputs)

Our LIVE-CODE programs range in size from just a few lines of code to substantial examples containing hundreds of lines of code. Each program is followed by a window containing the outputs produced when the program is run, so students can confirm that the programs run as expected. Relating outputs to the program statements that produce them is an excellent way to learn and to reinforce concepts. Our programs demonstrate the diverse features of Java. The code is syntax colored, with Java keywords, comments and other program text each appearing in different colors. This facilitates reading the code—students will especially appreciate the syntax coloring when they read the larger programs.

816 Illustrations/Figures

An abundance of charts, tables, line drawings, programs and program outputs is 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 additional types of UML diagrams throughout our optional OOD/UML ATM case study.

481 Programming Tips

We include programming tips to help students focus on important aspects of program development. We highlight these tips in the form of *Good Programming Practices*, *Common Programming Errors*, *Error-Prevention Tips*, *Look-and-Feel Observations*, *Performance Tips*, *Portability Tips* and *Software Engineering Observations*. These tips and practices represent the best we have gleaned from a combined six decades of programming and teaching experience. One of our students—a mathematics major—told us that she feels this approach is like the highlighting of axioms, theorems and corollaries in mathematics books; it provides a basis on which to build good software.



Good Programming Practices

Good Programming Practices are tips for writing clear programs. These techniques help students produce programs that are more readable, self-documenting and easier to maintain.



Common Programming Errors

Students who are new to programming (or a programming language) tend to make certain errors frequently. Focusing on these Common Programming Errors reduces the likelihood that students will make the same mistakes and shortens long lines outside instructors' offices during office hours!



Error-Prevention Tips

When we first designed this “tip type,” we thought we would use it strictly to tell people how to test and debug Java programs. In fact, many of the tips describe aspects of Java that reduce the likelihood of “bugs” and thus simplify the testing and debugging processes.



Look-and-Feel Observations

Look-and-Feel Observations highlight graphical user interface conventions. These observations help students design their own graphical user interfaces in conformance with industry norms.



Performance Tips

In our experience, teaching students to write clear and understandable programs is by far the most important goal for a first programming course. But students want to write the programs that run the fastest, use the least memory, require the smallest number of keystrokes, or dazzle in other nifty ways. Students really care about performance. They want to know what they can do to “turbo charge” their programs. So we highlight opportunities for improving program performance—making programs run faster or minimizing the amount of memory that they occupy.



Portability Tips

One of Java’s “claims to fame” is “universal” portability, so some programmers assume that an application written in Java will automatically be “perfectly” portable across all Java platforms. Unfortunately, this is not always the case. We include Portability Tips to help students write portable code and to provide insights on how Java achieves its high degree of portability.



Software Engineering Observations

The object-oriented programming paradigm requires a complete rethinking about the way we build software. Java is an effective language for performing good software engineering. The Software Engineering Observations highlight architectural and design issues that affect the construction of software systems, especially large-scale systems. Much of what the student learns here will be useful in upper-level courses and in industry as the student begins to work with large, complex, real-world systems.

Wrap-Up Section

New in this edition, each chapter ends with a brief “wrap-up” section that recaps the topics that were presented. Each section also helps the student transition to the next chapter.

Summary (1,303 Summary Bullets)

Each chapter ends with additional pedagogical devices. We present a thorough, bullet-liststyle summary of the chapter. On average, there are 45 summary bullets per chapter. This helps the students review and reinforce key concepts.

Terminology (2,388 Terms)

We include an alphabetized list of the important terms defined in each chapter—again, for further reinforcement. There is an average of 82 terms per chapter. Each term also appears in the index, and the defining occurrence of each term is highlighted in the index with a **blue, bold** page number so the student can locate the definitions of terms quickly.

701 Self-Review Exercises and Answers (Count Includes Separate Parts)

Extensive self-review exercises and answers are included for self-study. This gives the student a chance to build confidence with the material and prepare for the regular exercises. We encourage students to do all the self-review exercises and check their answers.

874 Exercises (Count Includes Separate Parts)

Each chapter concludes with a set of exercises, including simple recall of important terminology and concepts; writing individual Java statements; writing small portions of Java methods and classes; writing complete Java methods, classes, applications and applets; and writing major term projects. The large number of exercises across a wide variety of areas enables instructors to tailor their courses to the unique needs of their classes and to vary course assignments each semester. Instructors can use these exercises to form homework assignments, short quizzes and/or major examinations. The solutions for the vast majority of the exercises are included on the *Instructor's Resource CD (IRCD)*, which is available only to instructors through their Prentice Hall representatives^①. [NOTE: Please do not write to us requesting the Instructor's CD. Distribution of this ancillary is limited strictly to college instructors teaching from the book. Instructors may obtain the solutions manual only from their Prentice Hall representatives.] Students will have access to approximately half the exercises in the book in the free, Web-based *Cyber Classroom* which will be available in Spring 2005. For more information about the availability of the *Cyber Classroom*, please visit www.deitel.com or sign up for the free *Deitel® Buzz Online* e-mail newsletter at www.deitel.com/newsletter/subscribe.html.

Approximately 7150 Index Entries

We have included an extensive index at the back of the book. This helps students find terms or concepts by keyword. The Index is useful to people reading the book for the first time and is especially useful to practicing programmers who use the book as a reference.

"Double Indexing" of Java LIVE-CODE Examples

Java How to Program, 6/e has 256 live-code examples and 874 exercises (including parts). We have double indexed each of the live-code examples and most of the more substantial exercises. For every source-code program in the book, we indexed the figure caption both alphabetically and as a subindex item under "Examples." This makes it easier to find examples using particular features. The more substantial exercises are also indexed both alphabetically and as subindex items under "Exercises."

Tour of the Book

In this section, we take a tour of the many capabilities of Java that we explore in *Java How to Program, 6/e*. Figure 1 illustrates the dependencies among the chapters. We recommend studying these topics in the order indicated by the arrows, though other orders are possible. This book is widely used in all levels of Java programming courses. Search the Web for "syllabus," "Java" and "Deitel" to find syllabi used with recent editions this book.

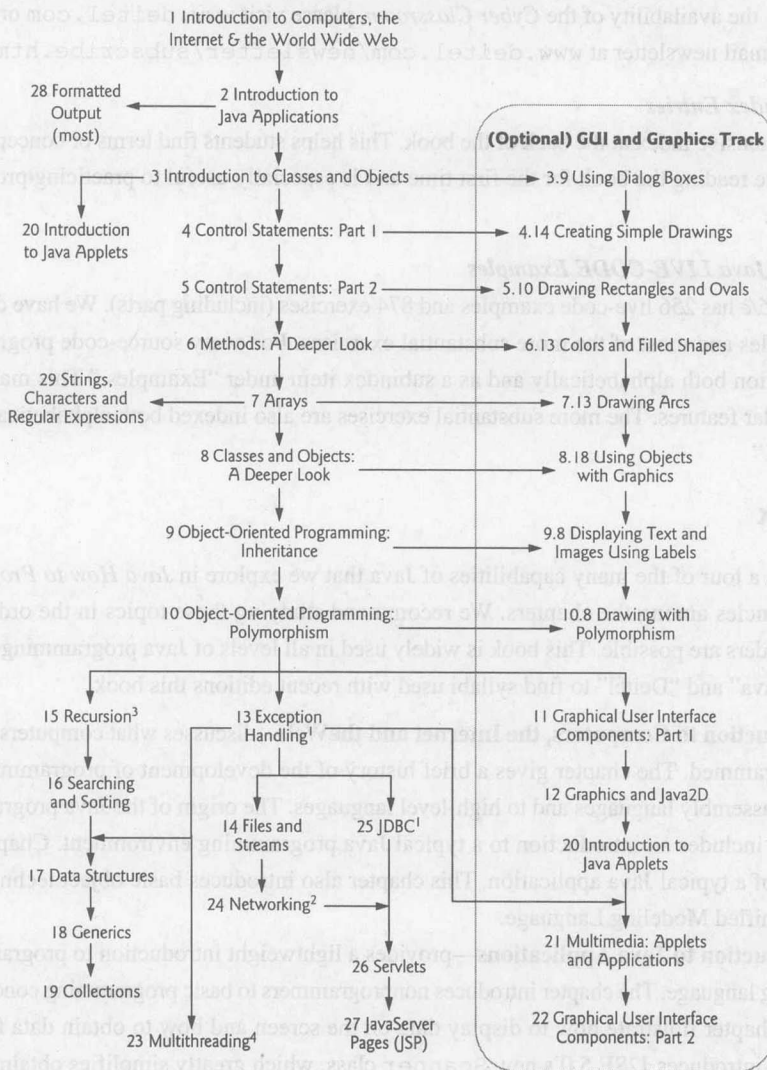
Chapter 1—Introduction to Computers, the Internet and the Web—discusses what computers are, how they work and how they are programmed. The chapter gives a brief history of the development of programming languages from machine languages, to assembly languages and to high-level languages. The origin of the Java programming language is discussed. The chapter includes an introduction to a typical Java programming environment. Chapter 1 walks readers through a "test drive" of a typical Java application. This chapter also introduces basic object technology concepts and terminology and the Unified Modeling Language.

Chapter 2—Introduction to Java Applications—provides a lightweight introduction to programming applications in the Java programming language. The chapter introduces nonprogrammers to basic programming concepts and constructs. The programs in this chapter illustrate how to display data on the screen and how to obtain data from the user at the keyboard. This chapter introduces J2SE 5.0's new `Scanner` class, which greatly simplifies obtaining user input. This chapter also introduces some of J2SE 5.0's new formatted output capabilities with method `System.out.printf`. Chapter 2 ends with detailed treatments of decision making and arithmetic operations.

Chapter 3—Introduction to Classes and Objects—introduces classes, objects, methods, constructors and instance variables using five simple real-world examples. The first four of these begin our **case study on developing a grade-book class** that instructors can use to maintain student test scores. The first example presents a `GradeBook` class with one method that simply displays a welcome message when it is called. We show how to create an object of that class and

① 国内的教师可通过填写书后的“教学支持说明”信息获取。——编者注

call the method so that it displays the welcome message. The second example modifies the first by allowing the method to receive a course name as an argument and by displaying the name as part of the welcome message. The third example illustrates storing the course name in a `GradeBook` object. For this version of the class, we also show how to set the course name and obtain the course name using methods. The fourth example demonstrates how the data in a `GradeBook` object can be initialized when the object is created—the initialization is performed by the class’s constructor. The last example introduces floating-point numbers in the context of a bank account class that maintains a customer’s balance. The chapter describes how to declare a class and use it to create an object and then discusses how to declare methods in a class to implement the class’s behaviors, how to declare instance variables in a class to implement the class’s attributes and how to call an object’s methods to make them perform their tasks. Chapter 3 explains the differences between instance variables of a class and local variables of a method, how to use a constructor to ensure that an object’s data is initialized when the object is created, and the differences between primitive and reference types.



1. Dependent on Chapter 11 for GUI used in one example.
2. Dependent on Chapter 20 for one example that uses an applet. The two large case studies at the end of this chapter each depend on Chapter 22 for GUI and Chapter 23 for multithreading.
3. Dependent on Chapters 11 and 12 for GUI and graphics used in one example.
4. Dependent on Chapter 11 for GUI used in one example. Dependent on Chapters 18 and 19 for one example.

Fig. 1 Flowchart illustrating the dependencies among chapters in *Java How to Program, 6/e*.

Chapter 4—Control Statements: Part 1—focuses on the program-development process. The chapter discusses how to take a problem statement and develop a working Java program from it, including performing intermediate steps