HZ BOOKS
华章教育

PEARSON

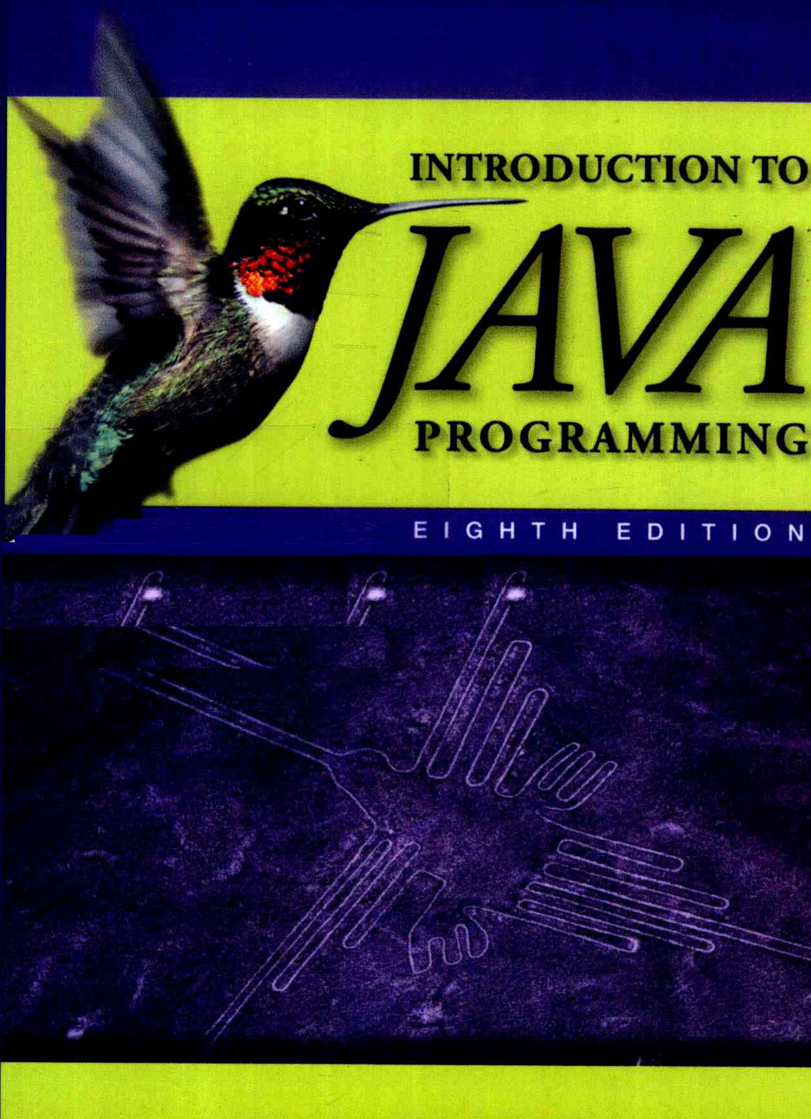
经典原版书库

Java语言程序设计

基础篇

（美）Y. Daniel Liang 著
阿姆斯特朗亚特兰大州立大学

（英文版·第8版）



机械工业出版社
China Machine Press

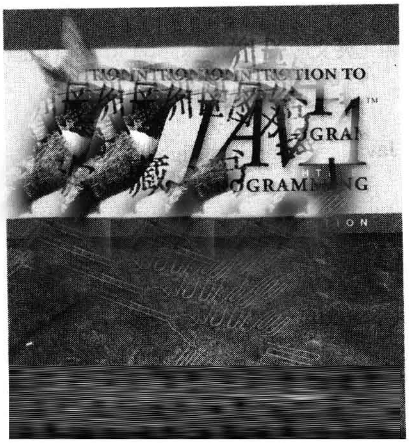
经典原版书库

Java语言程序设计

基础篇

（英文版・第8版）

Introduction to Java Programming (Eighth Edition)



（美）Y. Daniel Liang 著
阿姆斯特朗亚特兰大州立大学

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2011-5827

图书在版编目（CIP）数据

Java语言程序设计：基础篇（英文版 · 第8版）/（美）梁（Liang, Y. D.）著. —北京：机械工业出版社，2011.11

（经典原版书库）

书名原文：Introduction to Java Programming, Eighth Edition

ISBN 978-7-111-36122-0

Ⅰ. J…　Ⅱ. 梁…　Ⅲ. JAVA语言–程序设计–高等学校–教材–英文　Ⅳ. TP312

中国版本图书馆CIP数据核字（2011）第208084号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：迟振春
北京京师印务有限公司印刷
2012年1月第1版第1次印刷
170mm × 242mm · 45.75印张
标准书号：ISBN 978-7-111-36122-0
定价：89.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：(010) 88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

HZ BOOKS
华章教育
华章科技图书出版中心

PREFACE

This book uses the fundamentals-first approach and teaches programming concepts and techniques in a problem-driven way.

fundamentals-first

The fundamentals-first approach introduces basic programming concepts and techniques before objects and classes. My own experience, confirmed by the experiences of many colleagues, demonstrates that new programmers in order to succeed must learn basic logic and fundamental programming techniques such as loops and stepwise refinement. The fundamental concepts and techniques of loops, methods, and arrays are the foundation for programming. Building the foundation prepares students to learn object-oriented programming, GUI, database, and Web programming.

problem-driven

Problem-driven means focused on problem solving rather than syntax. We make introductory programming interesting by using interesting problems. The central thread of early chapters is on problem solving. Appropriate syntax and library are introduced to support the writing of a program for solving the problems. To support the teaching of programming in a problem-driven way, the book provides a wide variety of problems at various levels of difficulty to motivate students. In order to appeal to students in all majors, the problems cover many application areas in math, science, business, financials, gaming, animation, and multimedia.

Two Versions

comprehensive version

brief version

This *comprehensive version*[⊖] covers fundamentals of programming, object-oriented programming, GUI programming, algorithms and data structures, concurrency, networking, internationalization, advanced GUI, database, and Web programming. It is designed to prepare students to become proficient Java programmers. A *brief version* (Introduction to Java Programming, Brief Version, Eighth Edition) is available for a first course on programming, commonly known as CS1. The brief version contains the first 20 chapters of the comprehensive version.

What's New in This Edition?

This edition substantially improves *Introduction to Java Programming, Seventh Edition*. The major improvements are as follows:

- This edition is completely revised in every detail to enhance clarity, presentation, content, examples, and exercises.

complete revision

- In the examples and exercises, which are provided to motivate and stimulate student interest in programming, one-fifth of the problems are new.

new problems

- In the previous edition, console input was covered at the end of Chapter 2. The new edition introduces console input early in Chapter 2 so that students can write interactive programs early.

early console input

- The hand trace box is added for many programs to help novice students to read and trace programs.

hand trace box

- Single-dimensional arrays and multidimensional arrays are covered in two chapters to give instructors the flexibility to cover multidimensional arrays later.

multidimensional arrays

- The case study for the Sudoku problem has been moved to the Companion Website. A more pedagogically effective simple version of the Sudoku problem is presented instead.

Sudoku problem simplified

- The design of the API for Java GUI programming is an excellent example of how the object-oriented principle is applied. Students learn better with concrete and visual examples.

[⊖] 本书影印版将完全版分为基础篇和进阶篇。基础篇对应原书的第1～20章，进阶篇对应原书的第21～37章。——编辑注

basic GUI earlier

So, basic GUI now precedes the introduction of abstract classes and interfaces. The instructor, however, can still choose to cover abstract classes and interfaces before GUI.

exception handling earlier

- Exception handling is covered before abstract classes and interfaces so that students can build robust programs early. The instructor can still choose to cover exception handling later.

design guidelines

- Chapter 12, "Object-Oriented Design and Patterns," in the previous edition has been replaced by spreading the design guidelines and patterns into several chapters so that these topics can be covered in appropriate context.

sorting

- The chapter on sorting now follows right after the chapter on algorithm efficiency, so that students can immediately apply algorithm efficiency to sorting algorithms.

Java 2D

- A brand-new bonus Chapter 44 covers Java 2D.

new data structures chapters

- The coverage on data structures is expanded with new bonus chapters on AVL trees, splay trees, 2-4 trees, B-trees, and red-black trees, and hashing. So the book can be used for a full data structures course.

Learning Strategies

learn from mistakes

A programming course is quite different from other courses. In a programming course, you learn from examples, from practice, and *from mistakes*. You need to devote a lot of time to writing programs, testing them, and fixing errors.

programmatic solution

For first-time programmers, learning Java is like learning any high-level programming language. The fundamental point is to develop the critical skills of formulating programmatic solutions for real problems and translating them into programs using selection statements, loops, methods, and arrays.

object-oriented programming

Once you acquire the basic skills of writing programs using loops, methods, and arrays, you can begin to learn how to develop large programs and GUI programs using the object-oriented approach.

Java API

When you know how to program and you understand the concept of object-oriented programming, learning Java becomes a matter of learning the Java API. The Java API establishes a framework for programmers to develop applications using Java. You have to use the classes and interfaces in the API and follow their conventions and rules to create applications. The best way to learn the Java API is to imitate examples and do exercises.

Pedagogical Features

The book uses the following elements to get the most from the material:

- **Objectives** list what students should have learned from the chapter. This will help them determine whether they have met the objectives after completing the chapter.

- **Introduction** opens the discussion with representative problems to give the reader an overview of what to expect from the chapter.

- **Problems** carefully chosen and presented in an easy-to-follow style, teach problem solving and programming concepts. The book uses many small, simple, and stimulating examples to demonstrate important ideas.

- **Chapter Summary** reviews the important subjects that students should understand and remember. It helps them reinforce the key concepts they have learned in the chapter.

- **Review Questions** are grouped by sections to help students track their progress and evaluate their learning.

- **Programming Exercises** are grouped by sections to provide students with opportunities to apply on their own the new skills they have learned. The level of difficulty is rated as easy (no asterisk), moderate (*), hard (**), or challenging (***). The trick of learning programming is practice, practice, and practice. To that end, the book provides a great many exercises.

- **LiveLab** is a course assessment and management system. Students can submit programs online. The system automatically grades the programs/multiple-choice quizzes and gives students instant feedback. Instructors can create custom programming exercises and quizzes as well as use the system prebuilt exercises and quizzes.
- **Notes, Tips,** and **Cautions** are inserted throughout the text to offer valuable advice and insight on important aspects of program development.

Note

Provides additional information on the subject and reinforces important concepts.

Tip

Teaches good programming style and practice.

Caution

Helps students steer away from the pitfalls of programming errors.

Design Guide

Provides the guidelines for designing programs.

Flexible Chapter Orderings

The book is designed to provide flexible chapter orderings to enable GUI, exception handling, recursion, generics, and the Java Collections Framework to be covered earlier or later. The diagram on the next page shows the chapter dependencies.

Organization of the Book

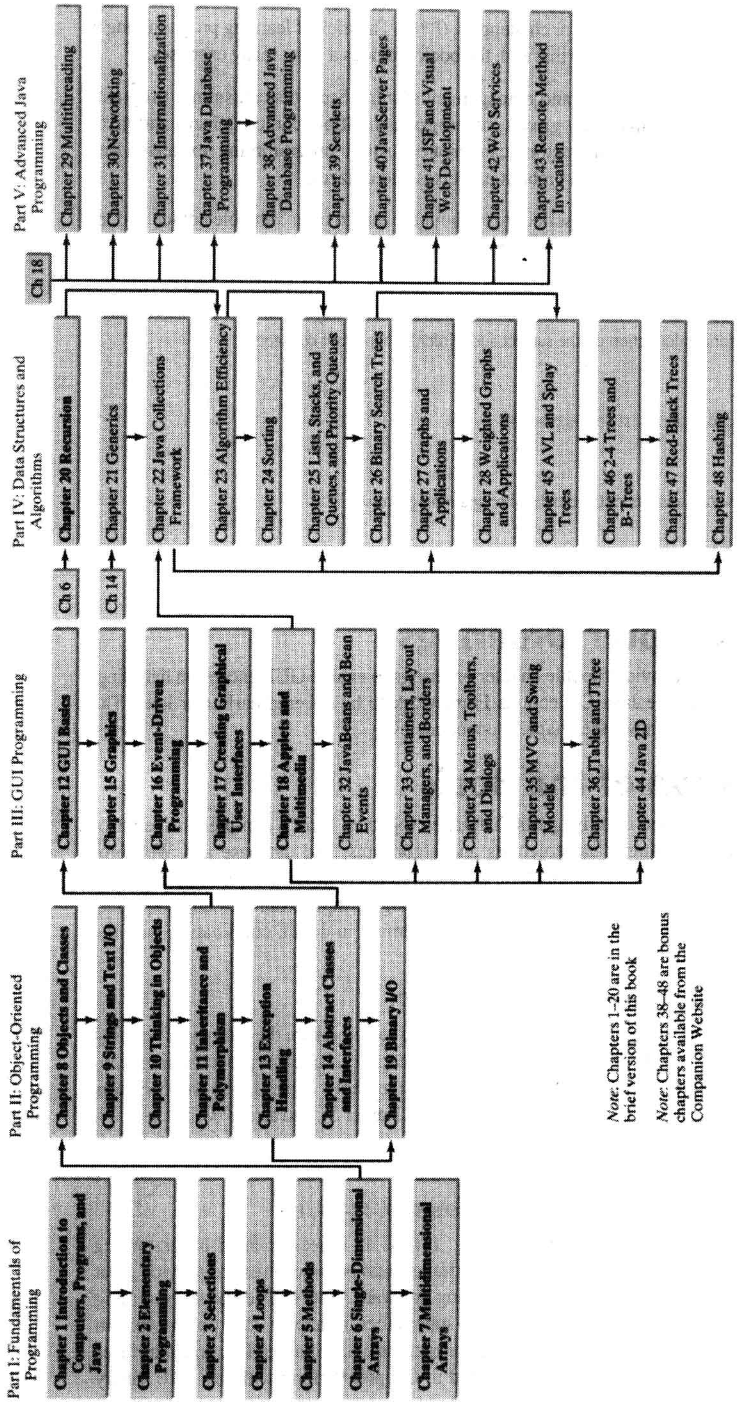
The chapters can be grouped into five parts that, taken together, form a comprehensive introduction to Java programming, data structures and algorithms, and database and Web programming. Because knowledge is cumulative, the early chapters provide the conceptual basis for understanding programming and guide students through simple examples and exercises; subsequent chapters progressively present Java programming in detail, culminating with the development of comprehensive Java applications.

Part I: Fundamentals of Programming (Chapters 1–7)

The first part of the book is a stepping stone, preparing you to embark on the journey of learning Java. You will begin to know Java (Chapter 1) and will learn fundamental programming techniques with primitive data types, variables, constants, assignments, expressions, and operators (Chapter 2), control statements (Chapters 3–4), methods (Chapter 5), and arrays (Chapters 6–7). After Chapter 6, you may jump to Chapter 20 to learn how to write recursive methods for solving inherently recursive problems.

Part II: Object-Oriented Programming (Chapters 8–11, 13–14, 19)

This part introduces object-oriented programming. Java is an object-oriented programming language that uses abstraction, encapsulation, inheritance, and polymorphism to provide great flexibility, modularity, and reusability in developing software. You will learn programming with objects and classes (Chapters 8–10), class inheritance (Chapter 11), polymorphism (Chapter 11), exception handling (Chapter 13), abstract classes (Chapter 14), and interfaces (Chapter 14). Processing strings will be introduced in Chapter 9 along with text I/O. Binary I/O is introduced in Chapter 19.


Part I: Fundamentals of Programming
Chapter 1 Introduction to Computers, Programs, and Java
Chapter 2 Elementary Programming
Chapter 3 Selections
Chapter 4 Loops
Chapter 5 Methods
Chapter 6 Single-Dimensional Arrays
Chapter 7 Multidimensional Arrays
Part II: Object-Oriented Programming
Chapter 8 Objects and Classes
Chapter 9 Strings and Text I/O
Chapter 10 Thinking in Objects
Chapter 11 Inheritance and Polymorphism
Chapter 13 Exception Handling
Chapter 14 Abstract Classes and Interfaces
Chapter 19 Binary I/O
Part III: GUI Programming
Chapter 12 GUI Basics
Chapter 15 Graphics
Chapter 16 Event-Driven Programming
Chapter 17 Creating Graphical User Interfaces
Chapter 18 Applets and Multimedia
Chapter 32 JavaBeans and Bean Events
Chapter 33 Containers, Layout Managers, and Borders
Chapter 34 Menus, Toolbars, and Dialogs
Chapter 35 MVC and Swing Models
Chapter 36 JTable and JTree
Chapter 44 Java 2D
Ch 6
Ch 14
Part IV: Data Structures and Algorithms
Chapter 20 Recursion
Chapter 21 Generics
Chapter 22 Java Collections Framework
Chapter 23 Algorithm Efficiency
Chapter 24 Sorting
Chapter 25 Lists, Stacks, and Queues, and Priority Queues
Chapter 26 Binary Search Trees
Chapter 27 Graphs and Applications
Chapter 28 Weighted Graphs and Applications
Chapter 45 AVL and Splay Trees
Chapter 46 2-4 Trees and B-Trees
Chapter 47 Red-Black Trees
Chapter 48 Hashing
Ch 18
Part V: Advanced Java Programming
Chapter 29 Multithreading
Chapter 30 Networking
Chapter 31 Internationalization
Chapter 37 Java Database Programming
Chapter 38 Advanced Java Database Programming
Chapter 39 Servlets
Chapter 40 JavaServer Pages
Chapter 41 JSF and Visual Web Development
Chapter 42 Web Services
Chapter 43 Remote Method Invocation

Note: Chapters 1–20 are in the brief version of this book

Note: Chapters 38–48 are bonus chapters available from the Companion Website

Part III: GUI Programming (Chapters 12, 15–18, 32–36, and 44)

This part introduces elementary Java GUI programming in Chapters 12 and 15–18 and advanced Java GUI programming in Chapters 32–36 and 44. Major topics include GUI basics (Chapter 12), drawing shapes (Chapter 15), event-driven programming (Chapter 16), creating graphical user interfaces (Chapter 17), and writing applets (Chapter 18). You will learn the architecture of Java GUI programming and use the GUI components to develop applications and applets from these elementary GUI chapters. The advanced GUI chapters introduce Java GUI programming in more depth and breadth. You will delve into JavaBeans and learn how to develop custom events and source components in Chapter 32, review and explore new containers, layout managers, and borders in Chapter 33, learn how to create GUI with menus, popup menus, toolbars, dialogs, and internal frames in Chapter 34, develop components using the MVC approach and explore the advanced Swing components **`JSpinner`**, **`JList`**, **`JComboBox`**, **`JSpinner`**, and **`JTable`**, and **`JTree`** in Chapters 35 and 36. Bonus Chapter 44 introduces Java 2D.

Part IV: Algorithms and Data Structures (Chapters 20–28, 45–48)

This part introduces the main subjects in a typical data structures course. Chapter 20 introduces recursion to write methods for solving inherently recursive problems. Chapter 21 introduces generics to improve software reliability. Chapter 22 introduces the Java Collection Framework, which defines a set of useful API for data structures. Chapter 23 introduces measurement of algorithm efficiency in order to choose an appropriate algorithm for applications. Chapter 24 introduces classic sorting algorithms. You will learn how to implement several classic data structures lists, queues, priority queues, binary search trees, AVL trees, splay trees, 2-4 trees, B-trees, and red-black trees in Chapters 25–26 and 45–47. Chapters 27 and 28 introduce graph applications. Chapter 48 introduces hashing.

Part V: Advanced Java Programming (Chapters 29–31, 37–43)

This part of the book is devoted to advanced Java programming. Chapter 29 treats the use of multithreading to make programs more responsive and interactive. Chapter 30 introduces how to write programs that talk with each other from different hosts over the Internet. Chapter 31 covers the use of internationalization support to develop projects for international audiences. Chapter 37 introduces the use of Java to develop database projects, Chapter 38 introduces advanced Java database programming, and Chapters 39 and 40 introduce how to use Java servlets and JSP to generate dynamic contents from Web servers. Chapter 41 introduces rapid Web application development using JavaServer Faces. Chapter 42 introduces Web services. Chapter 43 introduces remote method invocation.

Java Development Tools

IDE tutorials

You can use a text editor, such as the Windows Notepad or WordPad, to create Java programs and to compile and run the programs from the command window. You can also use a Java development tool, such as TextPad, NetBeans, or Eclipse. These tools support an integrated development environment (IDE) for rapidly developing Java programs. Editing, compiling, building, executing, and debugging programs are integrated in one graphical user interface. Using these tools effectively can greatly increase your programming productivity. TextPad is a primitive IDE tool. NetBeans and Eclipse are more sophisticated, but they are easy to use if you follow the tutorials. Tutorials on TextPad, NetBeans and Eclipse can be found in the supplements on the Companion Website.

LiveLab

This book is accompanied by an improved faster Web-based course assessment and management system. The system has three main components:

- **Automatic Grading System:** It can automatically grade programs from the text or created by instructors.
- **Quiz Creation/Submission/Grading System:** It enables instructors to create/modify quizzes that students can take and be graded upon automatically.
- **Tracking grades, attendance, etc:** The system enables the students to track grades and instructors, to view the grades of all students, and to track attendance.

The main features of the Automatic Grading System are as follows:

- Allows students to compile, run and submit exercises. (The system checks whether their program runs correctly—students can continue to run and resubmit the program before the due date.)
- Allows instructors to review submissions; run programs with instructor test cases; correct them; and provide feedback to students.
- Allows instructors to create/modify custom exercises, create public and secret test cases, assign exercises, and set due dates for the whole class or for individuals.
- All the exercises in the text can be assigned to students. Additionally, LiveLab provides extra exercises that are not printed in the text.
- Allows instructors to sort and filter all exercises and check grades (by time frame, student, and/or exercise).
- Allows instructors to delete students from the system.
- Allows students and instructors to track grades on exercises.

The main features of the Quiz System are as follows:

- Allows instructors to create/modify quizzes from test bank or a text file or to create complete new tests online.
- Allows instructors to assign the quizzes to students and set a due date and test time limit for the whole class or for individuals.
- Allows students and instructors to review submitted quizzes.
- Allows students and instructors to track grades on quizzes.

Video Notes are Pearson's new visual tool designed for teaching students key programming concepts and techniques. These short step-by-step videos demonstrate how to solve problems from design through coding. Video Notes allows for self-paced instruction with easy navigation including the ability to select, play, rewind, fast-forward, and stop within each Video Note exercise.

Video Note margin icons in your textbook let you know what a Video Notes video is available for a particular concept or homework problem.

Video Notes are free with the purchase of a new textbook. To purchase access to Video Notes, please go to **www.pearsonhighered.com/liang**.

Student Resource Materials

The student resources can be accessed through the Publisher's Web site (**www.pearsonhighered.com/liang**) and the Companion Web site (**www.cs.armstrong.edu/liang/intro8e**). The resources include:

- Answers to review questions
- Solutions to even-numbered programming exercises

- Source code for book examples
- Interactive self-test (organized by chapter sections)
- LiveLab
- Resource links
- Errata
- Video Notes
- Web Chapters

To access the Video Notes and Web Chapters, students must log onto www.pearsonhighered.com/liang and use the access card located in the front of the book to register and access the material. If there is no access card in the front of this textbook, students can purchase access by visiting www.pearsonhighered.com/liang and selecting *purchase access to premium content.*

Additional Supplements

The text covers the essential subjects. The supplements extend the text to introduce additional topics that might be of interest to readers. The supplements listed in this table are available from the Companion Web site.

Supplements on the Companion Web site

Part I General Supplements
A Glossary
B Installing and Configuring JDK
C Compiling and Running Java from the Command Window
D Java Coding Style Guidelines
E Creating Desktop Shortcuts for Java Applications on Windows
F Using Packages to Organize the Classes in the Text

Part II IDE Supplements
A TextPad Tutorial
B NetBeans Tutorial | One Page Startup Instruction
C Learning Java Effectively with NetBeans
D Eclipse Tutorial | One Page Startup Instruction
E Learning Java Effectively with Eclipse

Part III Java Supplements
A Java Characteristics
B Discussion on Operator and Operand Evaluations
C The & and | Operators
D Bitwise Operations
E Statement Labels with break and continue
F Enumerated Types
G Packages
H Regular Expressions
I Formatted Strings
J The Methods in the Object Class
K Hiding Data Fields and Static Methods
L Initialization Blocks
M Extended Discussions on Overriding Methods
N Design Patterns
O Text I/O Prior to JDK 1.5 (Reader and Writer Classes)
P Assertions
Q Packaging and Deploying Java Projects
R Java Web Start
S GridBagLayout | OverlayLayout | SpringLayout
T Networking Using Datagram Protocol
U Creating Internal Frames
V Pluggable Look and Feel
W UML Graphical Notations
X Testing Classes Using JUnit
Y JNI
Z The StringTokenizer Class

Part IV Database Supplements
A SQL Statements for Creating and Initializing Tables Used in the Book

B MySQL Tutorial
C Oracle Tutorial
D Microsoft Access Tutorial
E Introduction to Database Systems
F Relational Database Concept
G Database Design
H SQL Basics
I Advanced SQL

Part V Web Programming Supplements
A HTML and XHTML Tutorial
B CSS Tutorial
C XML
D Java and XML
E Tomcat Tutorial
F More Examples on JSF and Visual Web Development

Instructor Resource Materials

The instructor resources can be accessed through the Publisher's Web site (**www.pearsonhighered.com/liang**) and the Companion Web site (**www.cs.armstrong.edu/liang/intro8e**). For username and password information to the Liang 8e site, please contact your Pearson Representative.

The resources include:

- PowerPoint lecture slides with source code and run program capacity
- Instructor solutions manual
- Computerized test generator
- Sample exams using multiple choice and short answer questions, write and trace programs, and correcting programming errors.
- LiveLab
- Errata
- Video Notes
- Web Chapters

To access the Video Notes and Web Chapters, instructors must log onto **www.pearsonhighered.com/liang** and register.

Acknowledgments

I would like to thank Armstrong Atlantic State University for enabling me to teach what I write and for supporting me in writing what I teach. Teaching is the source of inspiration for continuing to improve the book. I am grateful to the instructors and students who have offered comments, suggestions, bug reports, and praise.

This book has been greatly enhanced thanks to outstanding reviews for this and previous editions. The reviewers are: Elizabeth Adams (James Madison University), Syed Ahmed (North Georgia College and State University), Omar Aldawud (Illinois Institute of Technology), Yang Ang (University of Wollongong, Australia), Kevin Bierre (Rochester Institute of Technology), David Champion (DeVry Institute), James Chegwidden (Tarrant County College), Anup Dargar (University of North Dakota), Charles Dierbach (Towson University), Frank Ducrest (University of Louisiana at Lafayette), Erica Eddy (University of Wisconsin at Parkside), Deena Engel (New York University), Henry A Etlinger (Rochester Institute of Technology), James Ten Eyck (Marist College), Olac Fuentes (University of Texas at El Paso), Harold Grossman (Clemson University), Barbara Guillot (Louisiana State University), Ron Hofman (Red River College, Canada), Stephen Hughes (Roanoke College), Vladan Jovanovic (Georgia Southern University), Edwin Kay (Lehigh University), Larry King (University of Texas at Dallas), Nana Kofi (Langara College, Canada), George Koutsogiannakis (Illinois

Institute of Technology), Roger Kraft (Purdue University at Calumet), Hong Lin (DeVry Institute), Dan Lipsa (Armstrong Atlantic State University), James Madison (Rensselaer Polytechnic Institute), Frank Malinowski (Darton College), Tim Margush (University of Akron), Debbie Masada (Sun Microsystems), Blayne Mayfield (Oklahoma State University), John McGrath (J.P. McGrath Consulting), Shyamal Mitra (University of Texas at Austin), Michel Mitri (James Madison University), Kenrick Mock (University of Alaska Anchorage), Jun Ni (University of Iowa), Benjamin Nystuen (University of Colorado at Colorado Springs), Maureen Opkins (CA State University, Long Beach), Gavin Osborne (University of Saskatchewan), Kevin Parker (Idaho State University), Dale Parson (Kutztown University), Mark Pendergast (Florida Gulf Coast University), Richard Povinelli (Marquette University), Roger Priebe (University of Texas at Austin), Mary Ann Pumphrey (De Anza Junior College), Pat Roth (Southern Polytechnic State University), Ronald F. Taylor (Wright State University), Carolyn Schauble (Colorado State University), David Scuse (University of Manitoba), Ashraf Shirani (San Jose State University), Daniel Spiegel (Kutztown University), Amr Sabry (Indiana University), Lixin Tao (Pace University), Russ Tront (Simon Fraser University), Deborah Trytten (University of Oklahoma), Kent Vidrine (George Washington University), and Bahram Zartoshty (California State University at Northridge).

It is a great pleasure, honor, and privilege to work with Pearson. I would like to thank Tracy Dunkelberger and her colleagues Marcia Horton, Margaret Waples, Erin Davis, Michael Hirsh, Matt Goldstein, Jake Warde, Melinda Haggerty, Allison Michael, Scott Disanno, Irwin Zucker, and their colleagues for organizing, producing, and promoting this project, and Robert Lentz for copy editing.

As always, I am indebted to my wife, Samantha, for her love, support, and encouragement.

Y. Daniel Liang
y.daniel.liang@gmail.com
www.cs.armstrong.edu/liang
www.pearsonhighered.com/liang

CONTENTS

[⊖] 自本章至第20章是基础篇内容。——编辑注