INTRODUCTION TO

JAVA

PROGRAMMING

COMPREHENSIVE VERSION

Eighth Edition

Y. Daniel Liang

Armstrong Atlantic State University

Prentice Hall
This book is dedicated to Dr. S. K. Dhall and Dr. S. Lakshmivarahan of the University of Oklahoma, who inspired me in teaching and research. Thank you for being my mentors and advisors.

To Samantha, Michael, and Michelle
This book uses the fundamentals-first approach and teaches programming concepts and techniques in a problem-driven way.

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 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

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.
- In the examples and exercises, which are provided to motivate and stimulate student interest in programming, one-fifth of the problems are new.
- 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.
- The hand trace box is added for many programs to help novice students to read and trace programs.
- Single-dimensional arrays and multidimensional arrays are covered in two chapters to give instructors the flexibility to cover multidimensional arrays later.
- 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.
- 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.
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 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.

- 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.

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

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

- 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**

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.

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.

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.

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 12 GUI Basics
Chapter 13 Exception Handling
Chapter 14 Abstract Classes and Interfaces
Chapter 15 Graphics
Chapter 16 Event-Driven Programming
Chapter 17 Creating Graphical User Interfaces
Chapter 18 Applets and Multimedia
Chapter 19 Binary I/O
Chapter 20 Recursion
Chapter 21 Generics
Chapter 22 Java Collections Framework

Part III: GUI Programming
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 29 Multithreading
Chapter 30 Networking
Chapter 31 Internationalization
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 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
Chapter 44 Java 2D
Chapter 45 AVL and Splay Trees
Chapter 46 2-4 Trees and B-Trees
Chapter 47 Red-Black Trees
Chapter 48 Hashing

Note: Chapters 1–20 are in the brief version of this book
Note: Chapters 38–48 are bonus chapters available from the Companion Website
Preface

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

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
Preface

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

<table>
<thead>
<tr>
<th>Part I General Supplements</th>
<th>Part II IDE Supplements</th>
<th>Part III Java Supplements</th>
<th>Part IV Database Supplements</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Glossary</td>
<td>A TextPad Tutorial</td>
<td>A Java Characteristics</td>
<td>A SQL Statements for Creating and Initializing Tables Used in the Book</td>
</tr>
<tr>
<td>B Installing and Configuring JDK</td>
<td>B NetBeans Tutorial</td>
<td>B Discussion on Operator and Operand Evaluations</td>
<td></td>
</tr>
<tr>
<td>C Compiling and Running Java from the Command Window</td>
<td>C Learning Java Effectively with NetBeans</td>
<td>C The &amp; and</td>
<td>Operators</td>
</tr>
<tr>
<td>D Java Coding Style Guidelines</td>
<td>D Eclipse Tutorial</td>
<td>D Bitwise Operations</td>
<td>D Initializing Blocks</td>
</tr>
<tr>
<td>E Creating Desktop Shortcuts for Java Applications on Windows</td>
<td>E Learning Java Effectively with Eclipse</td>
<td>E Statement Labels with break and continue</td>
<td>E Enumerated Types</td>
</tr>
<tr>
<td>F Using Packages to Organize the Classes in the Text</td>
<td>F TextPad Tutorial</td>
<td>F Enumerated Types</td>
<td>F Text I/O Prior to JDK 1.5 (Reader and Writer Classes)</td>
</tr>
<tr>
<td></td>
<td>B NetBeans Tutorial</td>
<td>G Packages</td>
<td>G Packages</td>
</tr>
<tr>
<td></td>
<td>C Learning Java Effectively with NetBeans</td>
<td>H Regular Expressions</td>
<td>H Regular Expressions</td>
</tr>
<tr>
<td></td>
<td>D Eclipse Tutorial</td>
<td>I Formatted Strings</td>
<td>I Formatted Strings</td>
</tr>
<tr>
<td></td>
<td>E Learning Java Effectively with Eclipse</td>
<td>J The Methods in the Object Class</td>
<td>J The Methods in the Object Class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K Hiding Data Fields and Static Methods</td>
<td>K Hiding Data Fields and Static Methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L Initialization Blocks</td>
<td>L Initialization Blocks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M Extended Discussions on Overriding Methods</td>
<td>M Extended Discussions on Overriding Methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N Design Patterns</td>
<td>N Design Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>O Text I/O Prior to JDK 1.5 (Reader and Writer Classes)</td>
<td>O Text I/O Prior to JDK 1.5 (Reader and Writer Classes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P Assertions</td>
<td>P Assertions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q Packaging and Deploying Java Projects</td>
<td>Q Packaging and Deploying Java Projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R Java Web Start</td>
<td>R Java Web Start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S GridBagLayout</td>
<td>S GridBagLayout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S OverlayLayout</td>
<td>S OverlayLayout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S SpringLayout</td>
<td>S SpringLayout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U Creating Internal Frames</td>
<td>U Creating Internal Frames</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V Pluggable Look and Feel</td>
<td>V Pluggable Look and Feel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W UML Graphical Notations</td>
<td>W UML Graphical Notations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X Testing Classes Using JUnit</td>
<td>X Testing Classes Using JUnit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y JNI</td>
<td>Y JNI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z The StringTokenizer Class</td>
<td>Z The StringTokenizer Class</td>
</tr>
</tbody>
</table>

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.
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
# BRIEF CONTENTS

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

## Appendices

A  Java Keywords  
B  The ASCII Character Set  
C  Operator Precedence Chart  
D  Java Modifiers  
E  Special Floating-Point Values  
F  Number Systems  

Index 1323
CONTENTS

**Chapter 1** Introduction to Computers, Programs, and Java

- 1.1 Introduction
- 1.2 What Is a Computer?
- 1.3 Programs
- 1.4 Operating Systems
- 1.5 Java, World Wide Web, and Beyond
- 1.6 The Java Language Specification, API, JDK, and IDE
- 1.7 A Simple Java Program
- 1.8 Creating, Compiling, and Executing a Java Program
- 1.9 (GUI) Displaying Text in a Message Dialog Box

**Chapter 2** Elementary Programming

- 2.1 Introduction
- 2.2 Writing Simple Programs
- 2.3 Reading Input from the Console
- 2.4 Identifiers
- 2.5 Variables
- 2.6 Assignment Statements and Assignment Expressions
- 2.7 Named Constants
- 2.8 Numeric Data Types and Operations
- 2.9 Problem: Displaying the Current Time
- 2.10 Shorthand Operators
- 2.11 Numeric Type Conversions
- 2.12 Problem: Computing Loan Payments
- 2.13 Character Data Type and Operations
- 2.14 Problem: Counting Monetary Units
- 2.15 The String Type
- 2.16 Programming Style and Documentation
- 2.17 Programming Errors
- 2.18 (GUI) Getting Input from Input Dialogs

**Chapter 3** Selections

- 3.1 Introduction
- 3.2 boolean Data Type
- 3.3 Problem: A Simple Math Learning Tool
- 3.4 if Statements
- 3.5 Problem: Guessing Birthdays
- 3.6 Two-Way if Statements
- 3.7 Nested if Statements
3.8 Common Errors in Selection Statements 81
3.9 Problem: An Improved Math Learning Tool 82
3.10 Problem: Computing Body Mass Index 84
3.11 Problem: Computing Taxes 85
3.12 Logical Operators 88
3.13 Problem: Determining Leap Year 90
3.14 Problem: Lottery 91
3.15 switch Statements 93
3.16 Conditional Expressions 95
3.17 Formatting Console Output 95
3.18 Operator Precedence and Associativity 97
3.19 (GUI) Confirmation Dialogs 98

Chapter 4  Loops 115
4.1 Introduction 116
4.2 The while Loop 116
4.3 The do-while Loop 124
4.4 The for Loop 126
4.5 Which Loop to Use? 128
4.6 Nested Loops 129
4.7 Minimizing Numeric Errors 130
4.8 Case Studies 131
4.9 Keywords break and continue 135
4.10 (GUI) Controlling a Loop with a Confirmation Dialog 139

Chapter 5  Methods 155
5.1 Introduction 156
5.2 Defining a Method 156
5.3 Calling a Method 158
5.4 void Method Example 160
5.5 Passing Parameters by Values 162
5.6 Modularizing Code 165
5.7 Problem: Converting Decimals to Hexadecimals 167
5.8 Overloading Methods 168
5.9 The Scope of Variables 171
5.10 The Math Class 172
5.11 Case Study: Generating Random Characters 175
5.12 Method Abstraction and Stepwise Refinement 176

Chapter 6  Single-Dimensional Arrays 197
6.1 Introduction 198
6.2 Array Basics 198
6.3 Problem: Lotto Numbers 204
6.4 Problem: Deck of Cards 206
6.5 Copying Arrays 208
Contents

6.6 Passing Arrays to Methods 209
6.7 Returning an Array from a Method 212
6.8 Variable-Length Argument Lists 215
6.9 Searching Arrays 216
6.10 Sorting Arrays 219
6.11 The Arrays Class 223

Chapter 7 Multidimensional Arrays 235
7.1 Introduction 236
7.2 Two-Dimensional Array Basics 236
7.3 Processing Two-Dimensional Arrays 238
7.4 Passing Two-Dimensional Arrays to Methods 240
7.5 Problem: Grading a Multiple-Choice Test 241
7.6 Problem: Finding a Closest Pair 242
7.7 Problem: Sudoku 244
7.8 Multidimensional Arrays 248

Chapter 8 Objects and Classes 263
8.1 Introduction 264
8.2 Defining Classes for Objects 264
8.3 Example: Defining Classes and Creating Objects 266
8.4 Constructing Objects Using Constructors 270
8.5 Accessing Objects via Reference Variables 270
8.6 Using Classes from the Java Library 274
8.7 Static Variables, Constants, and Methods 278
8.8 Visibility Modifiers 282
8.9 Data Field Encapsulation 283
8.10 Passing Objects to Methods 286
8.11 Array of Objects 287

Chapter 9 Strings and Text I/O 301
9.1 Introduction 302
9.2 The String Class 302
9.3 The Character Class 313
9.4 The StringBuilder/StringBuffer Class 315
9.5 Command-Line Arguments 320
9.6 The File Class 322
9.7 File Input and Output 325
9.8 (GUI) File Dialogs 329

Chapter 10 Thinking in Objects 343
10.1 Introduction 344
10.2 Immutable Objects and Classes 344
10.3 The Scope of Variables 345
10.4 The this Reference 346
10.5 Class Abstraction and Encapsulation 347
### Contents

10.6 Object-Oriented Thinking  
10.7 Object Composition  
10.8 Designing the Course Class  
10.9 Designing a Class for Stacks  
10.10 Designing the GuessDate Class  
10.11 Class Design Guidelines  

**Chapter 11** Inheritance and Polymorphism  
11.1 Introduction  
11.2 Superclasses and Subclasses  
11.3 Using the super Keyword  
11.4 Overriding Methods  
11.5 Overriding vs. Overloading  
11.6 The Object Class and Its toString() Method  
11.7 Polymorphism  
11.8 Dynamic Binding  
11.9 Casting Objects and the instanceof Operator  
11.10 The Object's equals() Method  
11.11 The ArrayList Class  
11.12 A Custom Stack Class  
11.13 The protected Data and Methods  
11.14 Preventing Extending and Overriding  

**Chapter 12** GUI Basics  
12.1 Introduction  
12.2 Swing vs. AWT  
12.3 The Java GUI API  
12.4 Frames  
12.5 Layout Managers  
12.6 Using Panels as Subcontainers  
12.7 The Color Class  
12.8 The Font Class  
12.9 Common Features of Swing GUI Components  
12.10 Image Icons  

**Chapter 13** Exception Handling  
13.1 Introduction  
13.2 Exception-Handling Overview  
13.3 Exception-Handling Advantages  
13.4 Exception Types  
13.5 More on Exception Handling  
13.6 The finally Clause  
13.7 When to Use Exceptions  
13.8 Rethrowing Exceptions  
13.9 Chained Exceptions  
13.10 Creating Custom Exception Classes  

# Contents

## Chapter 14  Abstract Classes and Interfaces  
457

### 14.1 Introduction  
458
### 14.2 Abstract Classes  
458
### 14.3 Example: Calendar and GregorianCalendar  
462
### 14.4 Interfaces  
465
### 14.5 Example: The Comparable Interface  
467
### 14.6 Example: The ActionListener Interface  
469
### 14.7 Example: The Cloneable Interface  
471
### 14.8 Interfaces vs. Abstract Classes  
473
### 14.9 Processing Primitive Data Type Values as Objects  
476
### 14.10 Sorting an Array of Objects  
479
### 14.11 Automatic Conversion between Primitive Types and Wrapper Class Types  
481
### 14.12 The BigInteger and BigDecimal Classes  
481
### 14.13 Case Study: The Rational Class  
482

## Chapter 15  Graphics  
497

### 15.1 Introduction  
498
### 15.2 Graphical Coordinate Systems  
498
### 15.3 The Graphics Class  
499
### 15.4 Drawing Strings, Lines, Rectangles, and Ovals  
501
### 15.5 Case Study: The FigurePanel Class  
502
### 15.6 Drawing Arcs  
506
### 15.7 Drawing Polygons and Polylines  
507
### 15.8 Centering a String Using the FontMetrics Class  
510
### 15.9 Case Study: The MessagePanel Class  
512
### 15.10 Case Study: The StillClock Class  
516
### 15.11 Displaying Images  
520
### 15.12 Case Study: The ImageViewer Class  
522

## Chapter 16  Event-Driven Programming  
533

### 16.1 Introduction  
534
### 16.2 Event and Event Source  
534
### 16.3 Listeners, Registrations, and Handling Events  
535
### 16.4 Inner Classes  
541
### 16.5 Anonymous Class Listeners  
542
### 16.6 Alternative Ways of Defining Listener Classes  
544
### 16.7 Problem: Loan Calculator  
547
### 16.8 Window Events  
549
### 16.9 Listener Interface Adapters  
551
### 16.10 Mouse Events  
552
### 16.11 Key Events  
555
### 16.12 Animation Using the Timer Class  
557

## Chapter 17  Creating Graphical User Interfaces  
571

### 17.1 Introduction  
572
### 17.2 Buttons  
572
### 17.3 Check Boxes  
578
17.4 Radio Buttons 581
17.5 Labels 583
17.6 Text Fields 584
17.7 Text Areas 586
17.8 Combo Boxes 590
17.9 Lists 593
17.10 Scroll Bars 596
17.11 Sliders 599
17.12 Creating Multiple Windows 602

Chapter 18 Applets and Multimedia 613
18.1 Introduction 614
18.2 Developing Applets 614
18.3 The HTML File and the <applet> Tag 615
18.4 Applet Security Restrictions 618
18.5 Enabling Applets to Run as Applications 618
18.6 Applet Life-Cycle Methods 620
18.7 Passing Strings to Applets 620
18.8 Case Study: Bouncing Ball 624
18.9 Case Study: TicTacToe 628
18.10 Locating Resources Using the URL Class 632
18.11 Playing Audio in Any Java Program 633
18.12 Case Study: Multimedia Animations 634

Chapter 19 Binary I/O 649
19.1 Introduction 650
19.2 How is I/O Handled in Java? 650
19.3 Text I/O vs. Binary I/O 650
19.4 Binary I/O Classes 652
19.5 Problem: Copying Files 660
19.6 Object I/O 662
19.7 Random-Access Files 666

Chapter 20 Recursion 677
20.1 Introduction 678
20.2 Problem: Computing Factorials 678
20.3 Problem: Computing Fibonacci Numbers 681
20.4 Problem Solving Using Recursion 683
20.5 Recursive Helper Methods 684
20.6 Problem: Finding the Directory Size 687
20.7 Problem: Towers of Hanoi 688
20.8 Problem: Fractals 692
20.9 Problem: Eight Queens 695
20.10 Recursion vs. Iteration 697
20.11 Tail Recursion 697
### Contents

25.3 Array Lists 825  
25.4 Linked Lists 830  
25.5 Variations of Linked Lists 842  
25.6 Stacks and Queues 843  
25.7 Priority Queues 846  
25.8 Case Study: Evaluating Expressions 847  

**Chapter 26** Binary Search Trees 857  
26.1 Introduction 858  
26.2 Binary Search Trees 858  
26.3 Deleting Elements in a BST 870  
26.4 Tree Visualization 876  
26.5 Iterators 879  
26.6 Case Study: Data Compression 881  

**Chapter 27** Graphs and Applications 891  
27.1 Introduction 892  
27.2 Basic Graph Terminologies 893  
27.3 Representing Graphs 894  
27.4 Modeling Graphs 898  
27.5 Graph Visualization 909  
27.6 Graph Traversals 911  
27.7 Depth-First Search (DFS) 912  
27.8 Breadth-First Search (BFS) 916  
27.9 Case Study: The Nine Tail Problem 919  
27.10 Case Study: The Knight’s Tour Problem 923  

**Chapter 28** Weighted Graphs and Applications 939  
28.1 Introduction 940  
28.2 Representing Weighted Graphs 940  
28.3 The WeightedGraph Class 942  
28.4 Minimum Spanning Trees 949  
28.5 Finding Shortest Paths 955  
28.6 Case Study: The Weighted Nine Tail Problem 962  

**Chapter 29** Multithreading 971  
29.1 Introduction 972  
29.2 Thread Concepts 972  
29.3 Creating Tasks and Threads 972  
29.4 The Thread Class 975  
29.5 Example: Flashing Text 978  
29.6 GUI Event Dispatch Thread 979  
29.7 Case Study: Clock with Audio 980  
29.8 Thread Pools 983
xxii  Contents

29.9 Thread Synchronization 985
29.10 Synchronization Using Locks 989
29.11 Cooperation among Threads 991
29.12 Case Study: Producer/Consumer 995
29.13 Blocking Queues 998
29.14 Semaphores 1000
29.15 Avoiding Deadlocks 1001
29.16 Thread States 1002
29.17 Synchronized Collections 1002
29.18 SwingWorker 1004
29.19 Displaying Progress Using JProgressBar 1007

Chapter 30 Networking 1017
30.1 Introduction 1018
30.2 Client/Server Computing 1018
30.3 The InetAddress Class 1025
30.4 Serving Multiple Clients 1026
30.5 Applet Clients 1029
30.6 Sending and Receiving Objects 1031
30.7 Retrieving Files from Web Servers 1036
30.8 JEditorPane 1039
30.9 Case Studies: Distributed TicTacToe Games 1041

Chapter 31 Internationalization 1057
31.1 Introduction 1058
31.2 The Locale Class 1058
31.3 Displaying Date and Time 1060
31.4 Formatting Numbers 1071
31.5 Resource Bundles 1077
31.6 Character Encoding 1084

Chapter 32 JavaBeans and Bean Events 1091
32.1 Introduction 1092
32.2 JavaBeans 1092
32.3 Bean Properties 1093
32.4 Java Event Model Review 1094
32.5 Creating Custom Source Components 1097
32.6 Creating Custom Event Sets 1101

Chapter 33 Containers, Layout Managers, and Borders 1111
33.1 Introduction 1112
33.2 Swing Container Structures 1112
33.3 Layout Managers 1114
33.4 Creating Custom Layout Managers 1123
Chapter 34 Menus, Toolbars, and Dialogs

34.1 Introduction 1150
34.2 Menus 1150
34.3 Popup Menus 1156
34.4 JToolBar 1158
34.5 Processing Actions Using the Action Interface 1160
34.6 JOptionPane Dialogs 1164
34.7 Creating Custom Dialogs 1171
34.8 JColorChooser 1174
34.9 JFileChooser 1176

Chapter 35 MVC and Swing Models

35.1 Introduction 1188
35.2 MVC 1188
35.3 MVC Variations 1194
35.4 Swing Model-View-Controller Architecture 1195
35.5 JSpinner 1196
35.6 Spinner Models and Editors 1198
35.7 JList and its Models 1205
35.8 List Models 1209
35.9 List Cell Renderer 1212
35.10 JComboBox and its Models 1216

Chapter 36 JTable and JTree

36.1 Introduction 1226
36.2 JTable 1226
36.3 Table Models and Table Column Models 1231
36.4 Auto Sort and Filtering 1235
36.5 Case Study: Modifying Rows and Columns 1237
36.6 Table Renderers and Editors 1242
36.7 Custom Table Renderers and Editors 1245
36.8 Table Model Events 1247
36.9 JTree 1250
36.10 TreeModel and DefaultTreeModel 1254
36.11 TreeNode, MutableTreeNode, and DefaultMutableTreeNode 1256
36.12 TreePath and TreeSelectionModel 1259
36.13 Case Study: Modifying Trees 1262
36.14 Tree Node Rendering and Editing 1265
36.15 Tree Events 1267
# Contents

## Chapter 37 Java Database Programming

37.1 Introduction 1274  
37.2 Relational Database Systems 1274  
37.3 SQL 1278  
37.4 JDBC 1286  
37.5 PreparedStatement 1295  
37.6 CallableStatement 1297  
37.7 Retrieving Metadata 1300

A detailed table of contents for the Web chapters is available on the companion Web site:

### Chapter 38 Advanced Java Database Programming 38-1  
### Chapter 39 Servlets 39-1  
### Chapter 40 JavaServer Pages 40-1  
### Chapter 41 JSF and Visual Web Development 41-1  
### Chapter 42 Web Services 42-1  
### Chapter 43 Remote Method Invocation 43-1  
### Chapter 44 Java 2D 44-1  
### Chapter 45 AVL Trees and Splay Trees 45-1  
### Chapter 46 2-4 Trees and B-Trees 46-1  
### Chapter 47 Red-Black Trees 47-1  
### Chapter 48 Hashing 48-1

## APPENDIXES

- Appendix A Java Keywords 1309  
- Appendix B The ASCII Character Set 1312  
- Appendix C Operator Precedence Chart 1314  
- Appendix D Java Modifiers 1316  
- Appendix E Special Floating-Point Values 1318  
- Appendix F Number Systems 1319  

## INDEX 1323