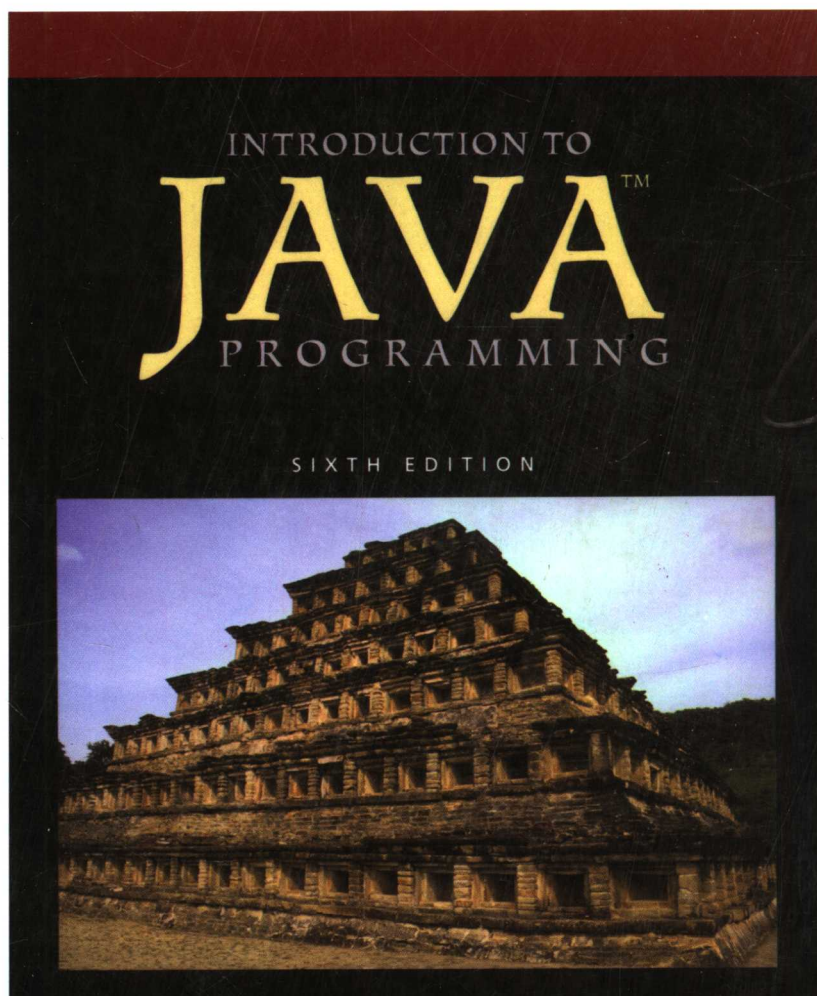


Java 语言程序设计

基础篇

(英文版·第6版)



(美) Y. Daniel Liang 著

经典原版书库

Java语言程序设计

基础篇

(英文版·第6版)

Introduction to Java Programming

(Sixth Edition)

TP312/Y203-2

江苏工业学院图书馆
藏书章

(美) Y. Daniel Liang 著



机械工业出版社
China Machine Press

English reprint edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Introduction to Java Programming, Sixth Edition* (ISBN 0-13-222158-6) by Y. Daniel Liang, Copyright © 2007, 2004, 2002, 2001.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-6511

图书在版编目（CIP）数据

Java语言程序设计 基础篇（英文版·第6版）/（美）梁（Liang, Y. D.）著.—北京：机械工业出版社，2008.2

（经典原版书库）

书名原文：Introduction to Java Programming, Sixth Edition

ISBN 978-7-111-23367-1

I. J… II. 梁… III. JAVA语言—程序设计—教材—英文 IV. TP312

中国版本图书馆CIP数据核字（2008）第009292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

北京京北制版厂印刷 · 新华书店北京发行所发行

2008年2月第1版第1次印刷

170mm × 242mm · 42印张

标准书号：ISBN 978-7-111-23367-1

定价：66.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

PREFACE

Welcome to *Introduction to Java Programming, Sixth Edition*. This edition is a substantial improvement on the previous edition in respect to clarity, content, presentation, code listings, and exercises, thanks to comments and suggestions by instructors and students. Overall, it is a great leap forward. We invite you to take a close look and be the judge.

Versions

The book is available in two versions:

- The fundamentals first version (Chapters 1–19).[⊖]
- The comprehensive version (Chapters 1–36).[⊕]

The following diagram summarizes the contents of the comprehensive version:

Introduction to Java Programming, Sixth Edition, Comprehensive Version	
Part 1 Fundamentals of Programming Chapter 1 Introduction to Computers, Programs, and Java Chapter 2 Primitive Data Types and Operations Chapter 3 Selection Statements Chapter 4 Loops Chapter 5 Methods Chapter 6 Arrays	Part 5 Data Structures Chapter 20 Lists, Stacks, Queues, Trees, and Heaps Chapter 21 Generics Chapter 22 Java Collections Framework Chapter 23 Algorithm Efficiency and Sorting
Part 2 Object-Oriented Programming Chapter 7 Objects and Classes Chapter 8 Strings and Text I/O Chapter 9 Inheritance and Polymorphism Chapter 10 Abstract Classes and Interfaces Chapter 11 Object-Oriented Design	Part 6 Concurrency, Networking, and Internationalization Chapter 24 Multithreading Chapter 25 Networking Chapter 26 Internationalization
Part 3 GUI Programming Chapter 12 Getting Started with GUI Programming Chapter 13 Graphics Chapter 14 Event-Driven Programming Chapter 15 Creating User Interfaces Chapter 16 Applets and Multimedia	Part 7 Advanced GUI Programming Chapter 27 JavaBeans and Bean Events Chapter 28 Containers, Layout Managers, and Borders Chapter 29 Menus, Toolbars, Dialogs, and Internal Frames Chapter 30 MVC and Swing Models Chapter 31 JTable and JTree
Part 4 Exception Handling, I/O, and Recursion Chapter 17 Exceptions and Assertions Chapter 18 Binary I/O Chapter 19 Recursion	Part 8 Web Programming Chapter 32 Java Database Programming Chapter 33 Advanced Java Database Programming Chapter 34 Servlets Chapter 35 JavaServer Pages Chapter 36 Remote Method Invocations

fundamentals first

Fundamentals First introduces the fundamentals of programming, problem-solving, object-oriented programming, and GUI programming. This version is suitable for an introductory course on problem-solving and object-oriented programming.

comprehensive version

The *Comprehensive Version* contains all the chapters in the fundamentals first version. Additionally, it covers data structures, networking, internationalization, advanced GUI programming, and Web programming.

⊖ 本部分对应影印版的基础篇。——编辑注

⊕ 原书完全版的影印版分为基础篇和进阶篇。本次出版的是基础篇（第1~19章）。——编辑注

Teaching Strategies

Both imperative and OOP are important programming paradigms with distinct advantages for certain applications. Some programs should be developed using the imperative approach and others are better developed using the object-oriented approach. Today's students need to know both paradigms and use them effectively. This book introduces both imperative and OOP paradigms. Students will learn when and how to apply these two paradigms effectively.

There are several strategies in teaching Java. This book adopts the *fundamentals-first* strategy, proceeding at a steady pace through all the necessary and important basic concepts, then moving to object-oriented programming, and then to the use of the object-oriented approach to build interesting GUI applications and applets with exception handling, and advanced features.

My own experience, confirmed by the experience of many colleagues, demonstrates that learning basic logic and *fundamental programming techniques* like loops and step-wise refinement is essential for new programmers to succeed. Students who cannot write code in procedural programming are not able to learn object-oriented programming. A good introduction on primitive data types, control statements, methods, and arrays prepares students to learn object-oriented programming.

The fundamentals-first approach reinforces object-oriented programming by first presenting the procedural solutions and demonstrating how they can be improved using the object-oriented approach. Students can learn when and how to apply OOP effectively.

At every SIGCSE (Computer Science Education) conference prior to 2005, the object-early approach was trumpeted and the voice for the fundamentals-first approach was muted. This changed when some former proponents of object-early began to air their frustrations and declared that object-early was a failure. This book is fundamentals-first and *object-right*. OOP is introduced right after fundamental programming techniques are covered. Many instructors of this book, from research universities to community colleges, have embraced the approach and have succeeded.

Programming isn't just syntax, classes, or objects. It is really *problem solving*. Loops, methods, and arrays are fundamental techniques for problem solving. From fundamental programming techniques to object-oriented programming, there are many layers of abstraction. Classes are simply a layer of abstraction. Applying the concept of abstraction in the design and implementation of software projects is the key to developing software. The overriding objective of the book, therefore, is to teach students to use many layers of abstraction in solving problems and to see problems in small and in large. The examples and exercises throughout the book center on problem solving and foster the concept of developing reusable methods and classes and using them to create practical projects.

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 in learning programming is to develop the critical skills of formulating *programmatic solutions* for real problems and translating them into programs using selection statements, loops, and methods.

Once you acquire the basic skills of writing programs using loops, methods, and arrays, you can begin to learn *object-oriented programming*. You will learn how to develop object-oriented software using class encapsulation and class inheritance.

Once 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. The following diagram highlights the API covered in the book.

imperative and OOP

fundamentals first

fundamental programming techniques

using OOP effectively

object-early failed?
object-right

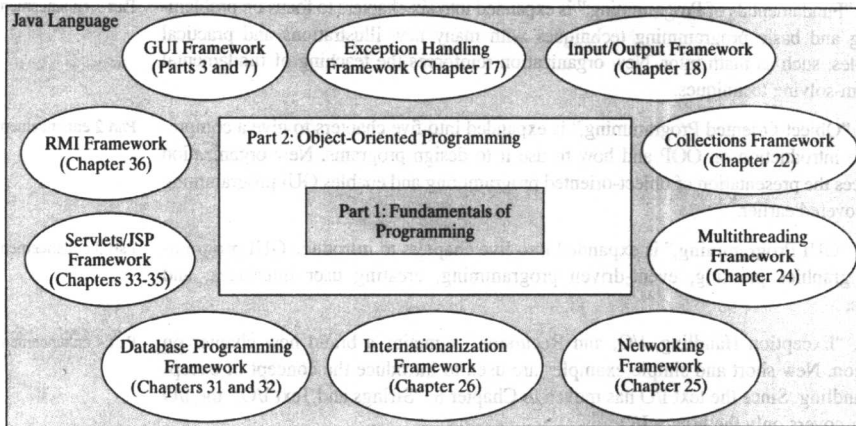
problem solving

practice

programmatic solution

object-oriented programming

Java API



What's New in This Edition?

This edition substantially enhances *Introduction to Java Programming, Fifth Edition*. The major changes are as follows:

- syntax coloring

 - All the Java code in the book now uses syntax coloring for keywords, literals, and comments. This makes the code more readable and understandable.
- full Java 5 update

 - The Java 5 features are fully integrated in this edition. (Note: the fifth edition treats Java 5 features in separate optional sections.)
- complete revision

 - The book is completely revised in every detail to enhance clarity, content, presentation, and exercises. All code samples and listings have been shortened and revised to be more compelling and to keep students even more engaged.
- GUI options in Part 2

 - Optional GUI sections are provided at the end of Chapters 7–10 in Part 2, “Object-Oriented Programming,” while a complete introduction of GUI programming is presented in Part 3, “GUI Programming,” GUI components are excellent examples to demonstrate OOP. The optional GUI sections can be used as additional examples for OOP.
- early interesting programs

 - Students can now write short, interesting, graphical game programs starting from Chapter 2.
- JOptionPane** or **Scanner**

 - Chapter 2, “Primitive Data Types and Operations,” introduces the use of **JOptionPane** and **Scanner** to receive input. Students can use either **JOptionPane** or **Scanner** for obtaining input.
- new recursion chapter

 - The discussion of recursion is expanded to form Chapter 19, “Recursion,” a brand-new chapter. It can be covered after Chapter 5, “Methods.”
- early text I/O

 - High-level text I/O using the **Scanner** and **PrintWriter** classes is introduced in Chapter 8 along with strings. Binary I/O is covered in Chapter 18 “Binary I/O.”
- early **ArrayList**

 - **ArrayList** is introduced early in Chapter 9, “Inheritance and Polymorphism.”
- UML exercises

 - New exercises are provided in Part 2, “Object-Oriented Programming,” for achieving three objectives: (1) design and draw UML for classes; (2) implement classes from the UML; (3) use classes to develop applications.
- extensive supplements

 - Extensive supplements (e.g., installing and configuring JDK, IDE tutorials, design patterns, rapid GUI development, database design, SQL) are provided for instructors to customize a course.


vi Preface

- Part 1, “Fundamentals of Programming,” is expanded into six chapters to focus on problem-solving and basic programming techniques with many new illustrations and practical examples, such as math tutor. New organization reinforces the teaching of fundamental problem-solving techniques. Part 1 enhancement
- Part 2, “Object-Oriented Programming,” is expanded into five chapters to give a comprehensive introduction on OOP and how to use it to design programs. New organization enhances the presentation of object-oriented programming and enables GUI programming to be covered earlier. Part 2 enhancement
- Part 3, “GUI Programming,” is expanded into five chapters to introduce GUI programming, graphics painting, event-driven programming, creating user interfaces, and applets. Part 3 enhancement
- Part 4, “Exception Handling, I/O, and Recursion,” contains a brand new chapter on recursion. New short and simple examples are used to introduce the concept of exception handling. Since the text I/O has moved to Chapter 8, “Strings and Text I/O,” the I/O chapter covers only the binary I/O. Part 4 enhancements
- Part 5, “Data Structures,” is expanded to cover data structure design and implementation (array list, linked list, stack, queue, heap, priority queue, and binary tree), generics, Java Collections Framework, and algorithm efficiencies and sorting. Part 5 enhancement
- Part 6, “Concurrency, Networking, and Internationalization,” is updated to cover Java 5 thread pooling, locks, and semaphores. Part 6 enhancement
- Part 7, “Advanced GUI Programming,” is expanded into five chapters with short, simple new examples to teach complex subjects. For example, a new example is used to demonstrate how to develop source components. The MVC architecture is introduced along with the Swing models. Part 7 enhancement
- Part 8, “Web Programming,” is expanded into five chapters. Advanced database programming is in a separate chapter and may be skipped. Several new examples are presented to introduce the JSP. Part 8 enhancement

Pedagogical Features

The philosophy of the Liang Java Series is *teaching by example and learning by doing*. Basic features are explained by example so that you can learn by doing. The book uses the following elements to get the most from the material:

teaching by example
learning by doing

- **Objectives** list what students should have learned from the chapter. This will help them to determine whether they have met the objectives after completing the chapter.
- **Introduction** opens the discussion with a brief overview of what to expect from the chapter.
- **Code Listings** are used to teach programming concepts. Syntax coloring makes the code easier to follow.
- **Chapter Summary** reviews the important subjects that students should understand and remember. It helps them to reinforce the key concepts they have learned in the chapter.
- **Optional Sections** cover nonessential but valuable features. Instructors may choose to include or skip an optional section, or cover it later. The section headers of optional sections are marked by .
- **Review Questions** are grouped by sections to help students track their progress and evaluate their learning.
- **Margin Notes** are featured throughout the book to emphasize key terms and important concepts.

- **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.
- **Interactive Self-Test** lets students test their knowledge interactively online. The Self-Test is accessible from the Companion Website. It provides more than one thousand multiple-choice questions organized by sections in each chapter. The Instructor Resource Website contains the quiz generator with additional multiple-choice questions.
- **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.

Flexible Chapter Orderings

The book provides flexible chapter orderings to enable GUI, exception handling, generics, and the Java Collections Framework to be covered earlier. Three common alternative orderings are shown as follows:

GUI Early

Chapter 9	Inheritance and Polymorphism
Chapter 12	GUI Basics
Chapter 13	Graphics
Chapter 10	Abstract Classes and Interfaces
Chapter 14	Event-Driven Programming
Chapter 15	Creating User Interfaces
Chapter 16	Applets and Multimedia

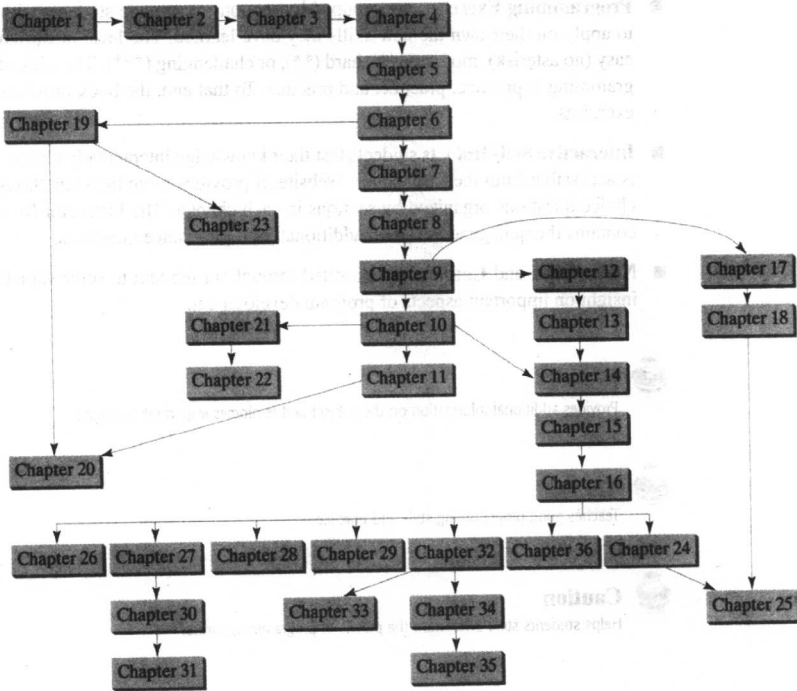
Exception Early

Chapter 9	Inheritance and Polymorphism
Chapter 17	Exceptions and Assertions

Data Structures Early

Chapter 9	Inheritance and Polymorphism
Chapter 10	Abstract Classes and Interfaces
Chapter 20	Lists, Stacks, Queues, Trees, and Heaps
Chapter 21	Generics
Chapter 22	Java Collections Framework

Many of the chapters after Chapter 16 can be covered in any order. The following diagram shows the chapter dependencies.



Note

Some of the optional examples and exercises in later chapters may be dependent on earlier chapters. In such cases the examples and exercises can be omitted. For example, Chapter 17 has an example that uses GUI components from Chapter 15. If you have not covered Chapter 15, these examples and exercises can be skipped.

Java Development Tools

You can use a text editor, such as the Windows Notepad or WordPad, to create Java programs, and compile and run the programs from the command window. You can also use a Java development tool, such as TextPad, JBuilder, 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 will greatly increase your programming productivity. TextPad is a primitive IDE tool. JBuilder, NetBeans, and Eclipse are more sophisticated, but they are easy to use if you follow the tutorials. Tutorials on TextPad, JBuilder, NetBeans, and Eclipse will be found in the supplements on the Companion Website.

Companion Website

The Companion Website at www.prenhall.com/liang features a host of resources for students and instructors, including the following:

- Answers to review questions
- Solutions to programming exercises
- Source code for the code listings in the book

- Links to software and Web resources
- The **Online Quiz Generator** enables instructors to generate quizzes as well as assign students to take quizzes online with the results emailed to the instructor. This can be done in a closed lab or at home. Instructors may assign a set of 10, 15, or 20 questions and specify that the questions to be randomly generated. Go to www.prenhall.com/liang to access the Online Quiz.

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 are available from Companion Website.

Instructor Resource Center

The Instructor Resource Center, accessible from www.prenhall.com/liang, contains the following resources:

- Over 2000 Microsoft PowerPoint lecture slides. They bring the illustrations from the text to life in the classroom. Featuring full-color, syntax-highlighted source code and the ability to run programs live without leaving the slides and trace program execution using animation.
- Sample exams. In general, each exam has four parts:
 1. Multiple-choice questions or short-answer questions (most of these are different from the questions in the self-test on the Companion Website)
 2. Correct programming errors
 3. Trace programs
 4. Write programs
- Solutions to all the exercises (available to instructors only). Students will have access to the solutions of even-numbered exercises.
- UML Diagram Solutions to all the UML exercises (for instructors only). Students are provided solutions to even-numbered exercises.
- TestGen, a test bank of over 2000 programming questions.
- Online quiz. (Students can take the online quiz for each chapter and a quiz report will be sent to the instructor.)
- Ten sample exams, each with multiple choice or short-answer questions. They test how students correct programming errors, test programs, and write their own programs.
- Online Homework and Assessment. GOAL (Gradience Online Accelerated Learning) is a sophisticated online homework tool that can be purchased and used alongside the textbook. Multiple choice problems based on the concept of a “root question” and challenging programming labs are designed to save instructors time grading programming assignments and to deliver meaningful feedback to students immediately. All code is tested against a real data set. Contact your local Pearson Prentice Hall sales representative to learn more about GOAL.

Some readers have requested the materials from the Instructor Resource Website. Please understand that these are for instructors only. Such requests will not be answered.

Acknowledgments

I would like to thank Ray Greenlaw and my colleagues at Armstrong Atlantic State University for enabling me to teach what I write and for supporting me in writing what I teach. Teaching

x Preface

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 was greatly enhanced thanks to outstanding reviews. The reviewers for the previous editions were:

Yang Ang	University of Wollongong (Australia)
David Champion	DeVry Institute
James Chegwiddden	Tarrant County College
Harold Grossman	Clemson University
Ron Hofman	Red River College (Canada)
Hong Lin	DeVry Institute
Dan Lipsa	Armstrong Atlantic State University
Vladan Jovanovic	Georgia Southern University
Larry King	University of Texas at Dallas
Nana Kofi	Langara College (Canada)
Roger Kraft	Purdue University at Calumet
Debbie Masada	Sun Microsystems
Blayne Mayfield	Oklahoma State University
Michel Mitri	James Madison University
Kenrick Mock	University of Alaska Anchorage
Jun Ni	University of Iowa
Gavin Osborne	University of Saskatchewan
Kevin Parker	Idaho State University
Mary Ann Pumphrey	De Anza Junior College
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
Lixin Tao	Pace University
Russ Tront	Simon Fraser University
Deborah Trytten	University of Oklahoma
Kent Vidrine	George Washington University
Bahram Zartoshty	California State University at Northridge

The reviewers for this edition are:

Kevin Bierre	Rochester Institute of Technology
James Chegwiddden	Tarrant County College
Charles Dierbach	Towson University
Deena Engel	New York University
Vladan Jovanovic	Georgia Southern University
Frank Malinowski	Aelera Corporation
John McGrath	J.P. McGrath Consulting
Shyamal Mitra	University of Texas at Austin
Benjamin Nystuen	University of Colorado at Colorado
Roger Priebe	University of Texas at Austin

It is a great pleasure, honor, and privilege to work with Prentice Hall. I would like to thank Marcia Horton, Tracy Dunkelberger, Robin O'Brien, Christianna Lee, Jennifer Cappello, Vince O'Brien, Camille Trentacoste, Craig Little, Xiaohong Zhu, and their colleagues for organizing, producing, and promoting this project, and Robert Milch for copy editing.

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

Y. Daniel Liang
liang@armstrong.edu
www.cs.armstrong.edu/liang/intro6e

CONTENTS

PART 1 FUNDAMENTALS OF PROGRAMMING 1

Chapter 1 Introduction to Computers, Programs, and Java 3

1.1 Introduction	4
1.2 What Is a Computer?	4
1.3 Programs	7
1.4 Operating Systems	10
1.5 (Optional) Number Systems	11
1.6 Java, the World Wide Web, and Beyond	14
1.7 The Java Language Specification, API, JDK, and IDE	16
1.8 A Simple Java Program	17
1.9 Creating, Compiling, and Executing a Java Program	18
1.10 Anatomy of a Java Program	20
1.11 Displaying Text in a Message Dialog Box	22

Chapter 2 Primitive Data Types and Operations 27

2.1 Introduction	28
2.2 Writing Simple Programs	28
2.3 Identifiers	30
2.4 Variables	31
2.5 Assignment Statements and Assignment Expressions	32
2.6 Constants	33
2.7 Numeric Data Types and Operations	34
2.8 Numeric Type Conversions	40
2.9 Character Data Type and Operations	41
2.10 The String Type	44
2.11 Getting Input from Input Dialogs	45
2.12 Case Studies	46
2.13 Getting Input from the Console	52
2.14 Programming Style and Documentation	55
2.15 Programming Errors	57
2.16 Debugging	58

Chapter 3 Selection Statements 67

3.1 Introduction	68
3.2 boolean Data Type and Operations	68
3.3 if Statements	73
3.4 switch Statements	81

3.5 Conditional Expressions	83
3.6 Formatting Console Output and Strings	84
3.7 Operator Precedence and Associativity	86
3.8 Operand Evaluation Order	87

Chapter 4 Loops 95

4.1 Introduction	96
4.2 The while Loop	96
4.3 The do-while Loop	100
4.4 The for Loop	102
4.5 Which Loop to Use?	104
4.6 Nested Loops	105
4.7 Minimizing Numerical Errors	106
4.8 Case Studies	108
4.9 Keywords break and continue	113

Chapter 5 Methods 129

5.1 Introduction	130
5.2 Creating a Method	130
5.3 Calling a Method	131
5.4 void Method Example	134
5.5 Passing Parameters by Values	135
5.6 Overloading Methods	137
5.7 Case Study: Computing Taxes with Methods	139
5.8 The Scope of Variables	141
5.9 The Math Class	142
5.10 Case Study: Generating Random Characters	145
5.11 Method Abstraction and Stepwise Refinement	146
5.12 (Optional) Packages	154

Chapter 6 Arrays 169

6.1 Introduction	170
6.2 Array Basics	170
6.3 Copying Arrays	177
6.4 Passing Arrays to Methods	179
6.5 Returning an Array from a Method	181
6.6 (Optional) Variable-Length Argument Lists	185
6.7 Searching Arrays	186
6.8 Sorting Arrays	188

xii Contents

6.9	The Arrays Class	191	9.3	Using the super Keyword	307
6.10	Two-Dimensional Arrays	193	9.4	Overriding Methods	309
6.11	(Optional) Multidimensional Arrays	199	9.5	Overriding vs. Overloading	310
PART 2 ■ OBJECT-ORIENTED PROGRAMMING		211	9.6	The Object Class and Its toString() Method	310
Chapter 7 Objects and Classes		213	9.7	Polymorphism, Dynamic Binding, and Generic Programming	311
7.1	Introduction	214	9.8	Casting Objects and the instanceof Operator	313
7.2	Defining Classes for Objects	214	9.9	The ArrayList Class	315
7.3	Constructors	216	9.10	A Custom Stack Class	319
7.4	Accessing Objects via Reference Variables	216	9.11	The protected Data and Methods	320
7.5	Using Classes from the Java Library	222	9.12	The final Classes, Methods, and Variables	321
7.6	Static Variables, Constants, and Methods	224	9.13	(Optional) Methods in the Object Class	322
7.7	Visibility Modifiers	228	9.14	(Optional) Hiding Data Fields and Static Methods	325
7.8	Data Field Encapsulation	229	9.15	(Optional) Initialization Blocks	327
7.9	Immutable Objects and Classes	232	9.16	(Optional GUI) Inheriting GUI Components	330
7.10	Passing Objects to Methods	233	Chapter 10 Abstract Classes and Interfaces		341
7.11	The Scope of Variables	235	10.1	Introduction	342
7.12	The this Keyword	236	10.2	Abstract Classes	342
7.13	Array of Objects	237	10.3	The Calendar and GregorianCalendar Classes	346
7.14	Class Abstraction and Encapsulation	239	10.4	Interfaces	347
7.15	Case Study: The Loan Class	239	10.5	Processing Primitive Data Type Values as Objects	356
7.16	Case Study: The Course Class	243	10.6	Automatic Conversion Between Primitive Types and Wrapper Class Types	362
7.17	(Optional) Case Study: The StackOfIntegers Class	245	10.7	(Optional GUI) Handling GUI Events	363
7.18	(Optional GUI) Creating Windows	247	Chapter 11 Object-Oriented Design		371
Chapter 8 Strings and Text I/O		261	11.1	Introduction	372
8.1	Introduction	262	11.2	The Software Development Process	372
8.2	The String Class	262	11.3	Discovering Class Relationships	373
8.3	The Character Class	270	11.4	Case Study: Object-Oriented Design	377
8.4	The StringBuilder/StringBuffer Class	273	11.5	Case Study: The Rational Class	382
8.5	Command-Line Arguments	277	11.6	Class Design Guidelines	387
8.6	(Optional) Regular Expressions	279	11.7	(Optional) Modeling Dynamic Behavior Using Sequence Diagrams and Statecharts	390
8.7	The File Class	283	11.8	Framework-Based Programming Using Java API	392
8.8	Text I/O	286	PART 3 ■ GUI PROGRAMMING		397
8.9	(Optional GUI) Containers and Layout Managers	290	Chapter 12 Getting Started with GUI Programming		399
Chapter 9 Inheritance and Polymorphism		301	12.1	Introduction	400
9.1	Introduction	302	12.2	GUI Components	400
9.2	Superclasses and Subclasses	302			

12.3	The Java GUI API	401	15.7	Text Areas	506
12.4	Frames	403	15.8	Combo Boxes	509
12.5	Layout Managers	406	15.9	Lists	512
12.6	The Color Class	412	15.10	Scroll Bars	515
12.7	The Font Class	413	15.11	Sliders	518
12.8	Using Panels as Subcontainers	413	15.12	Creating Multiple Windows	521
12.9	Common Features of Swing GUI Components	415			
12.10	Image Icons	418	Chapter 16	Applets and Multimedia	535
Chapter 13	Graphics	425	16.1	Introduction	536
13.1	Introduction	426	16.2	The Applet Class	536
13.2	Graphical Coordinate Systems	426	16.3	The JApplet Class	538
13.3	The Graphics Class	426	16.4	The HTML File and the <applet> Tag	539
13.4	The paintComponent Method	429	16.5	Enabling Applets to Run as Applications	544
13.5	Drawing Graphics on Panels	430	16.6	Passing Strings to Applets	546
13.6	Drawing Strings, Lines, Rectangles, and Ovals	431	16.7	(Optional) Case Study: TicTacToe	550
13.7	Case Study: The FigurePanel Class	432	16.8	(Optional) Case Study: Bouncing Ball	554
13.8	Drawing Arcs	436	16.9	(Optional) Locating Resource Using the URL Class	557
13.9	The Polygon class and Drawing Polygons and Polylines	438	16.10	(Optional) Playing Audio	559
13.10	Centering a Display Using the FontMetrics Class	440	16.11	(Optional) Case Study: Multimedia Animations	560
13.11	Case Study: The MessagePanel Class	442	16.12	(Optional) Packaging and Deploying Java Projects	562
13.12	(Optional) Case Study: The StillClock Class	447			
13.13	(Optional) Displaying Images	451	PART 4	EXCEPTION HANDLING, I/O, AND RECURSION	575
13.14	(Optional) Case Study: The ImageViewer Class	452			
Chapter 14	Event-Driven Programming	463	Chapter 17	Exceptions and Assertions	577
14.1	Introduction	464	17.1	Introduction	578
14.2	Event and Event Source	465	17.2	Exception-Handling Overview	578
14.3	Listeners, Registrations, and Handling Events	465	17.3	Exceptions and Exception Types	580
14.4	Mouse Events	477	17.4	Understanding Exception Handling	582
14.5	Key Events	479	17.5	The finally Clause	589
14.6	(Optional) Animation Using the Timer Class	482	17.6	When to Use Exceptions	590
Chapter 15	Creating User Interfaces	491	17.7	Rethrowing Exceptions	591
15.1	Introduction	492	17.8	(Optional) Chained Exceptions	591
15.2	Buttons	492	17.9	(Optional) Creating Custom Exception Classes	592
15.3	Check Boxes	498	17.10	(Optional) Assertions	593
15.4	Radio Buttons	501	Chapter 18	Binary I/O	605
15.5	Labels	503	18.1	Introduction	606
15.6	Text Fields	504	18.2	How is I/O Handled in Java?	606

xiv Contents

18.3	Text I/O vs. Binary I/O	606	Chapter 22	Java Collections Framework	713
18.4	Binary I/O Classes	608	22.1	Introduction	714
18.5	Case Study: Copying File	615	22.2	The Collection Interface and the AbstractCollection Class	715
18.6	Object I/O	617	22.3	Sets	716
18.7	(Optional) Random Access Files	621	22.4	The Comparator Interface	721
18.8	(Optional) Case Study: Address Book	623	22.5	Lists	723
Chapter 19	Recursion	635	22.6	Static Methods for Lists and Collections	726
19.1	Introduction	636	22.7	The Vector and Stack Classes	729
19.2	Example: Factorials	636	22.8	Queues and Priority Queues	731
19.3	Example: Fibonacci Numbers	638	22.9	(Optional) Maps	733
19.4	Problem Solving Using Recursion	640	22.10	Singleton and Unmodifiable Collections and Maps	737
19.5	Recursive Helper Methods	641	Chapter 23	Algorithm Efficiency and Sorting	745
19.6	Tower of Hanoi	643	23.1	Introduction	746
19.7	Fractals	646	23.2	Estimating Algorithm Efficiency	746
19.8	Recursion versus Iteration	649	23.3	Bubble Sort	749
PART 5 [⊖]	DATA STRUCTURES	657	23.4	Merge Sort	751
Chapter 20	Lists, Stacks, Queues, Trees, and Heaps	659	23.5	Quick Sort	754
20.1	Introduction	660	23.6	Heap Sort	757
20.2	Lists	660	23.7	External Sort	761
20.3	Stacks and Queues	675	PART 6	CONCURRENCY, NETWORKING, AND INTERNATIONALIZATION	771
20.4	Binary Trees	677	Chapter 24	Multithreading	773
20.5	Heaps	683	24.1	Introduction	774
20.6	Priority Queues	688	24.2	Thread Concepts	774
Chapter 21	Generics	693	24.3	Creating Tasks and Threads	774
21.1	Introduction	694	24.4	The Thread Class	777
21.2	Motivations	694	24.5	Example: Flashing Text	780
21.3	Declaring Generic Classes and Interfaces	695	24.6	GUI Event Dispatcher Thread	781
21.4	Generic Methods	698	24.7	(Optional) Case Study: Clock with Audio	782
21.5	Raw Type and Backward Compatibility	699	24.8	Thread Pools	785
21.6	Wildcards	701	24.9	Thread Synchronization	787
21.7	Important Facts	703	24.10	(Optional) Synchronization Using Locks	790
21.8	Avoiding Unsafe Raw Types	703	24.11	(Optional) Cooperation Among Threads	792
21.9	(Optional) Case Study: Generic Matrix Class	705	24.12	(Optional) Case Study: Producer/Consumer	797
			24.13	(Optional) Blocking Queues	799
			24.14	(Optional) Semaphores	801

24.15	Avoiding Deadlocks	803	28.4	Creating Custom Layout Managers	938
24.16	Thread States	803	28.5	JScrollPane	943
24.17	Synchronized Collections	804	28.6	JTabbedPane	947
24.18	(Optional) JProgressBar	805	28.7	JSplitPane	949
Chapter 25 Networking		815	28.8	Swing Borders	952
25.1	Introduction	816	28.9	(Optional) Pluggable Look-and-Feel	960
25.2	Client/Server Computing	816	Chapter 29 Menus, Toolbars, Dialogs, and Internal Frames		
25.3	The InetAddress Class	822			967
25.4	Serving Multiple Clients	823	29.1	Introduction	968
25.5	Applet Clients	826	29.2	Menus	968
25.6	Sending and Receiving Objects	829	29.3	Popup Menus	974
25.7	Retrieving Files from Web Servers	834	29.4	JToolBar	976
25.8	JEditorPane	836	29.5	Processing Actions Using the Action Interface	978
25.9	(Optional) Case Studies: Distributed TicTacToe Games	838	29.6	JOptionPane Dialogs	982
25.10	(Optional) Datagram Socket	850	29.7	Creating Custom Dialogs	989
Chapter 26 Internationalization		861	29.8	JColorChooser	992
26.1	Introduction	862	29.9	JFileChooser	994
26.2	The Locale Class	862	29.10	(Optional) Creating Internal Frames	999
26.3	Displaying Date and Time	864	Chapter 30 MVC and Swing Models		
26.4	Formatting Numbers	875			1007
26.5	(Optional) Resource Bundles	881	30.1	Introduction	1008
26.6	(Optional) Character Encoding	888	30.2	MVC	1008
PART 7 * ADVANCED GUI PROGRAMMING		895	30.3	MVC Variations	1014
Chapter 27 JavaBeans and Bean Events		897	30.4	Swing Model-View-Controller Architecture	1015
27.1	Introduction	898	30.5	JSpinner	1016
27.2	JavaBeans	898	30.6	Spinner Models and Editors	1018
27.3	Bean Properties	899	30.7	JList	1024
27.4	Java Event Model Review	900	30.8	List Models	1029
27.5	Creating Custom Source Components	902	30.9	List Cell Renderer	1032
27.6	(Optional) Creating Custom Event Sets	906	30.10	JComboBox	1035
Chapter 28 Containers, Layout Managers, and Borders		917	Chapter 31 JTable and JTree		
28.1	Introduction	918			1045
28.2	Swing Container Structures	918	31.1	Introduction	1046
28.3	Layout Managers	920	31.2	JTable	1046
			31.3	Table Models and Table Column Models	1051
			31.4	Case Study: Modifying Rows and Columns	1055
			31.5	Table Renderers and Editors	1060
			31.6	Custom Table Renderers and Editors	1062
			31.7	Table Model Events	1065
			31.8	JTree	1068

xvi Contents

31.9	TreeModel and DefaultTreeModel	1073	35.4	JSP Scripting Constructs	1211
31.10	TreeNode, MutableTreeNode, and DefaultMutableTreeNode	1074	35.5	Predefined Variables	1213
31.11	TreePath and TreeSelectionMode	1077	35.6	JSP Directives	1215
31.12	Case Study: Modifying Trees	1080	35.7	Using JavaBeans in JSP	1219
31.13	Tree Node Rendering and Editing	1084	35.8	Getting and Setting Properties	1222
31.14	Tree Events	1085	35.9	Associating Properties with Input Parameters	1223
			35.10	Forwarding Requests from JavaServer Pages	1232
			35.11	Case Study: Browsing Database Tables	1232
PART 8	WEB PROGRAMMING	1091			
Chapter 32	Java Database Programming	1093	Chapter 36	Remote Method Invocations	1243
32.1	Introduction	1094	36.1	Introduction	1244
32.2	Relational Database Systems	1094	36.2	RMI Basics	1244
32.3	SQL	1098	36.3	Developing RMI Applications	1246
32.4	JDBC	1106	36.4	RMI vs. Socket-Level Programming	1252
32.5	PreparedStatement	1113	36.5	Developing Three-Tier Applications Using RMI	1253
32.6	Retrieving Metadata	1116	36.6	RMI Callbacks	1255
Chapter 33	Advanced Java Database Programming	1125			
33.1	Introduction	1126	APPENDIXES		1269
33.2	A Universal SQL Client	1126	Appendix A	Java Keywords	1271
33.3	Batch Processing	1130	Appendix B	The ASCII Character Set	1272
33.4	Scrollable and Updatable Result Set	1135	Appendix C	Operator Precedence Chart	1274
33.5	RowSet, JdbcRowSet, and CachedRowSet	1145	Appendix D	Java Modifiers	1276
33.6	Storing and Retrieving Images in JDBC	1150	Appendix E	Special Floating-Point Values	1278
Chapter 34	Servlets	1157	Appendix F	Bit Operations	1279
34.1	Introduction	1158			
34.2	HTML and Common Gateway Interface	1158	INDEX		1281
34.3	Creating and Running Servlets	1160			
34.4	The Servlet API	1164			
34.5	Creating Servlets	1168			
34.6	HTML Forms	1170			
34.7	Database Programming in Servlets	1176			
34.8	Session Tracking	1180			
34.9	Sending Images from Servlets	1195			
Chapter 35	JavaServer Pages	1209			
35.1	Introduction	1210			
35.2	A Simple JSP Page	1210			
35.3	How Is a JSP Page Processed?	1211			