

FIFTH EDITION

BIG JAVA

EARLY OBJECTS



Cay S. Horstmann

International Student Version

INTERNATIONAL STUDENT VERSION

BIG JAVA

Early Objects

Fifth Edition



CAY HORSTMANN

San Jose State University

WILEY

Copyright © 2014, 2010 John Wiley & Sons Singapore. Pte. Ltd.

Cover image © Ruddi Handen/iStockphoto

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

All rights reserved. **This book is authorized for sale in Europe, Asia, Africa and the Middle East only and may not be exported outside of these territories.** Exportation from or importation of this book to another region without the Publisher's authorization is illegal and is a violation of the Publisher's rights. The Publisher may take legal action to enforce its rights. The Publisher may recover damages and costs, including but not limited to lost profits and attorney's fees, in the event legal action is required.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, website www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, website <http://www.wiley.com/go/permissions>.

ISBN: 978-1-118-31877-5

Printed in Asia

10987654321

PREFACE

This book is an introduction to Java and computer programming that focuses on the essentials—and on effective learning. The book is designed to serve a wide range of student interests and abilities and is suitable for a first course in programming for computer scientists, engineers, and students in other disciplines. No prior programming experience is required, and only a modest amount of high school algebra is needed. Here are the key features of this book:

Start objects early, teach object orientation gradually.

In Chapter 2, students learn how to use objects and classes from the standard library. The chapter then shows the mechanics of implementing classes *from a given specification*. Students then use simple objects as they master branches, loops, and arrays. Object-oriented design starts in Chapter 7. This gradual approach allows students to use objects throughout their study of the core algorithmic topics, without teaching bad habits that must be un-learned later.

Guidance and worked examples help students succeed.

Beginning programmers often ask “How do I start? Now what do I do?” Of course, an activity as complex as programming cannot be reduced to cookbook-style instructions. However, step-by-step guidance is immensely helpful for building confidence and providing an outline for the task at hand. “Problem Solving” sections stress the importance of design and planning. “How To” guides help students with common programming tasks. Additional Worked Examples are available online.

Practice makes perfect.

Of course, programming students need to be able to implement nontrivial programs, but they first need to have the confidence that they can succeed. This book contains a substantial number of self-check questions at the end of each section. “Practice It” pointers suggest exercises to try after each section. And additional practice opportunities, including lab exercises and skill-oriented multiple-choice questions are available online.

A visual approach motivates the reader and eases navigation.

Photographs present visual analogies that explain the nature and behavior of computer concepts. Step-by-step figures illustrate complex program operations. Syntax boxes and example tables present a variety of typical and special cases in a compact format. It is easy to get the “lay of the land” by browsing the visuals, before focusing on the textual material.



Visual features help the reader with navigation.

Focus on the essentials while being technically accurate.

An encyclopedic coverage is not helpful for a beginning programmer, but neither is the opposite—reducing the material to a list of simplistic bullet points. In this book, the essentials are presented in digestible chunks, with separate notes that go deeper into good practices

or language features when the reader is ready for the additional information. You will not find artificial over-simplifications that give an illusion of knowledge.

Reinforce sound engineering practices.

A multitude of useful tips on software quality and common errors encourage the development of good programming habits. The optional testing track focuses on test-driven development, encouraging students to test their programs systematically.

Provide an optional graphics track.

Graphical shapes are splendid examples of objects. Many students enjoy writing programs that create drawings or use graphical user interfaces. If desired, these topics can be integrated into the course by using the materials at the end of Chapters 2 and 8, and Chapter 18.

New to This Edition

Problem Solving Strategies

This edition adds practical, step-by-step illustrations of techniques that can help students devise and evaluate solutions to programming problems. Introduced where they are most relevant, these strategies address barriers to success for many students. Strategies included are:

- Algorithm Design (with pseudocode)
- Tracing Objects
- First Do It By Hand (doing sample calculations by hand)
- Flowcharts
- Selecting Test Cases
- Hand-Tracing
- Storyboards
- Adapting Algorithms
- Discovering Algorithms by Manipulating Physical Objects
- Patterns for Object Data
- Thinking Recursively
- Estimating the Running Time of an Algorithm

Optional Science and Business Exercises

End-of-chapter exercises have been enhanced with problems from scientific and business domains. Designed to engage students, the exercises illustrate the value of programming in applied fields.

New and Reorganized Topics

All chapters were revised and enhanced to respond to user feedback and improve the flow of topics. Loop algorithms are now introduced explicitly in Chapter 5. Additional array algorithms are presented in Chapter 6 and incorporated into the problem-solving sections. Chapter 7 is more clearly focused on the design of a single class, whereas Chapter 11 deals with relationships between classes. The coverage of data structures has been completely reorganized. Chapter 14 covers the use of existing data structures. The implementation of linked lists and stacks is now in Chapter 15, and a greatly enhanced Chapter 16 covers binary search trees, red-black trees, and trees whose nodes have more than two children. New example tables, photographs, and exercises appear throughout the book.

A Tour of the Book

The book can be naturally grouped into four parts, as illustrated by Figure 1. The organization of chapters offers the same flexibility as the previous edition; dependencies among the chapters are also shown in the figure.

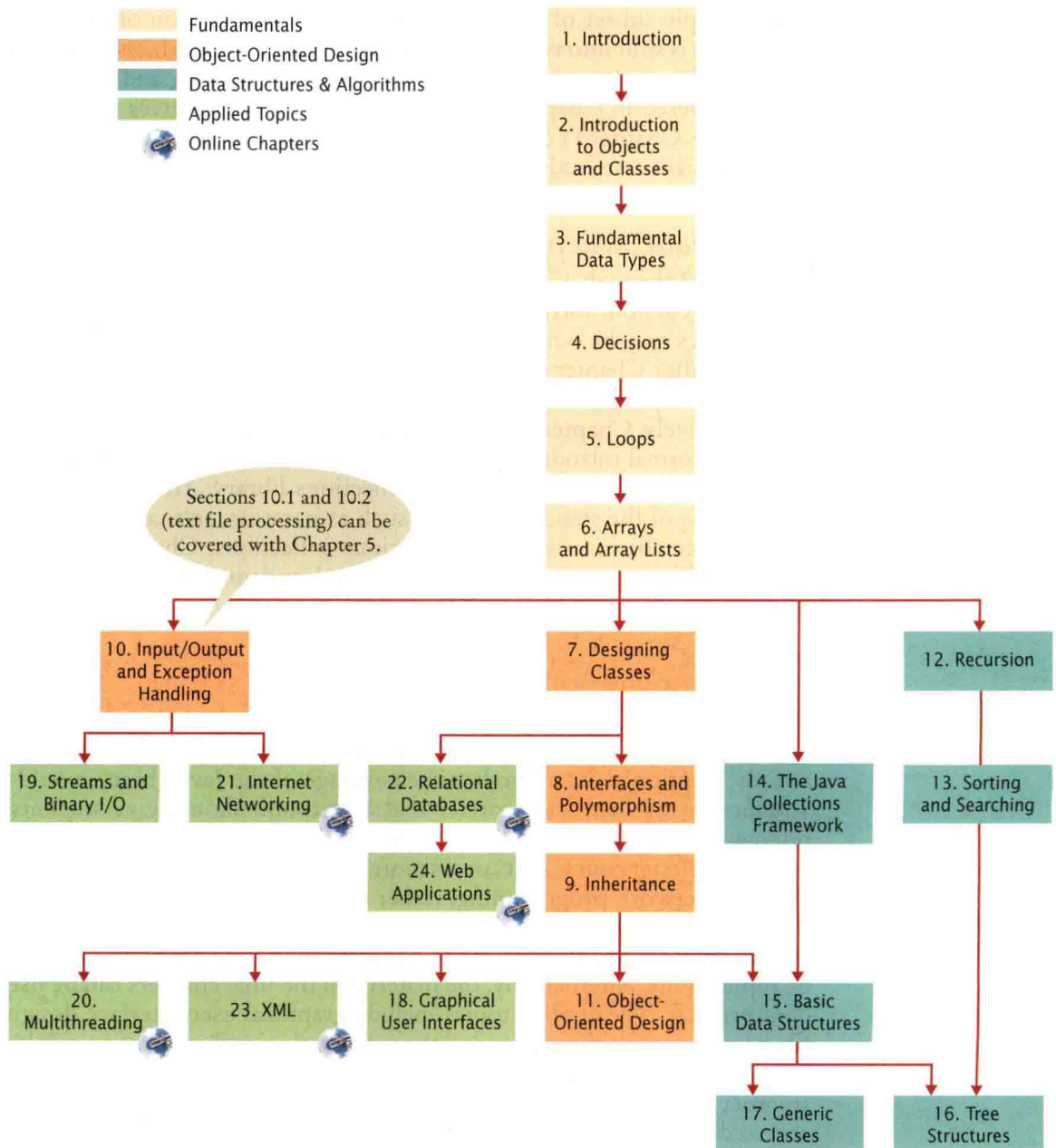


Figure 1 Chapter Dependencies

Part A: Fundamentals (Chapters 1–6)

Chapter 1 contains a brief introduction to computer science and Java programming. Chapter 2 shows how to manipulate objects of predefined classes, and how to build your own simple classes from given specifications. Fundamental data types, branches, loops, and arrays are covered in Chapters 3–6.

Part B: Object-Oriented Design (Chapters 7–11)

Chapter 7 takes up the subject of class design in a systematic fashion, and it introduces a very simple subset of the UML notation. The discussion of polymorphism and inheritance is split into two chapters. Chapter 8 covers interfaces and polymorphism, whereas Chapter 9 covers inheritance. Exception handling and basic file input/output are covered in Chapter 10. The exception hierarchy gives a useful example for inheritance. Chapter 11 contains an introduction to object-oriented design, including two significant case studies.

Part C: Data Structures and Algorithms (Chapters 12–17)

Chapters 12 through 17 contain an introduction to algorithms and data structures, covering recursion, sorting and searching, linked lists, binary trees, and hash tables. These topics may be outside the scope of a one-semester course, but can be covered as desired after Chapter 6 (see Figure 1). Recursion, in Chapter 12, starts with simple examples and progresses to meaningful applications that would be difficult to implement iteratively. Chapter 13 covers quadratic sorting algorithms as well as merge sort, with an informal introduction to big-Oh notation. Each data structure is presented in the context of the standard Java collections library. You will learn the essential abstractions of the standard library (such as iterators, sets, and maps) as well as the performance characteristics of the various collections. Chapter 17 introduces Java generics. This chapter is suitable for advanced students who want to implement their own generic classes and methods.

Part D: Applied Topics (Chapters 18–24)



Chapters 18 through 24 cover Java programming techniques that definitely go beyond a first course in Java (20–24 are on the book's companion site). Although, as already mentioned, a comprehensive coverage of the Java library would span many volumes, many instructors prefer that a textbook should give students additional reference material valuable beyond their first course. Some institutions also teach a second-semester course that covers more practical programming aspects such as database and network programming, rather than the more traditional in-depth material on data structures and algorithms. This book can be used in a two-semester course to give students an introduction to programming fundamentals and broad coverage of applications. Alternatively, the material in the final chapters can be useful for student projects. The applied topics include graphical user-interface design, advanced file handling, multithreading, and those technologies that are of particular interest to server-side programming: networking, databases, XML, and web applications. The Internet has made it possible to deploy many useful applications on servers, often accessed by nothing more than a browser. This server-centric approach to application development was in part made possible by the Java language and libraries, and today, much of the industrial use of Java is in server-side programming.

Appendices

Many instructors find it highly beneficial to require a consistent style for all assignments. If the style guide in Appendix I conflicts with instructor sentiment or local customs, however, it is available in electronic form so that it can be modified.

- | | |
|---|------------------------------------|
| A. The Basic Latin and Latin-1 Subsets of Unicode | F. Tool Summary |
| B. Java Operator Summary | G. Number Systems |
| C. Java Reserved Word Summary | H. UML Summary |
| D. The Java Library | I. Java Language Coding Guidelines |
| E. Java Syntax Summary | J. HTML Summary |

Web Resources

This book is complemented by a complete suite of online resources. Go to www.wiley.com/college/horstmann to visit the online companion sites, which include

- “CodeCheck,” a new online service currently in development by Cay Horstmann that students can use to check their homework assignments and to work on additional practice problems. Visit <http://horstmann.com/codecheck> to learn more and to try it out.
- Source code for all example programs in the book and in online examples.
- Worked Examples that apply the problem-solving steps in the book to other realistic examples.
- Animations of key concepts.
- Lab exercises that apply chapter concepts (with solutions for instructors only).
- Lecture presentation slides (for instructors only).
- Solutions to all review and programming exercises (for instructors only).
- A test bank that focuses on skills, not just terminology (for instructors only). This extensive set of multiple-choice questions can be used with a word processor or imported into a course management system.

A Walkthrough of the Learning Aids

The pedagogical elements in this book work together to focus on and reinforce key concepts and fundamental principles of programming, with additional tips and detail organized to support and deepen these fundamentals. For a tour of these elements, please visit the online companion site at www.wiley.com/horstmann and choose the “Walkthrough of Learning Aids”.

Pointers in the book describe what students will find on the Web.

WORKED EXAMPLE 5.3 A Sample Debugging Session
Learn how to find bugs in an algorithm for counting the syllables of a word. Go to wiley.com/go/javaexamples and download Worked Example 5.3.

FULL CODE EXAMPLE
Go to wiley.com/go/javacode to download a program that demonstrates variables and assignments.

COMMUNICATOR
mu·ni·ca·tor
com·mu·ni·c
communicatir
-es, or i

Acknowledgments

Many thanks to Beth Lang Golub, Don Fowley, Elizabeth Mills, Katherine Willis, Jenny Welter, Wendy Ashenberg, Lisa Gee, Kevin Holm, and Tim Lindner at John Wiley & Sons, and Vickie Piercey at Publishing Services for their help with this project. An especially deep acknowledgment and thanks goes to Cindy Johnson for her hard work, sound judgment, and amazing attention to detail.

I am grateful to Suchindran Chatterjee, *Arizona State University*, Jose Cordova, *University of Louisiana*, Udayan Das, *DeVry University*, James Johnson, Aaron Keen, *California Polytechnic State University San Luis Obispo*, Norm Krumpe, *Miami University Ohio*, Kathy Liszka, *University of Akron*, Kathleen O'Brien, *San Jose State University*, Donald Smith, *Columbia College*, Mark Thomas, *University of Cincinnati*, Laurie White, *Mercer University*, Brent Wilson, *George Fox University*, and David Woolbright, *Columbus State University*, for their excellent contributions to the supplementary materials.

Many thanks to the individuals who reviewed the manuscript for this edition, made valuable suggestions, and brought an embarrassingly large number of errors and omissions to my attention. They include:

Eric Aaron, <i>Wesleyan University</i>	Guy Helmer, <i>Iowa State University</i>	Bill Mongan, <i>Drexel University</i>
James Agnew, <i>Anne Arundel Community College</i>	Ed Holden, <i>Rochester Institute of Technology</i>	George Novacky, <i>University of Pittsburgh</i>
Greg Ballinger, <i>Miami Dade College</i>	Steven Janke, <i>Colorado College</i>	Mimi Opkins, <i>California State University Long Beach</i>
Jon Beck, <i>Truman State University</i>	Mark Jones, <i>Lock Haven University of Pennsylvania</i>	Derek Pao, <i>City University of Hong Kong</i>
Matt Boutell, <i>Rose-Hulman Institute of Technology</i>	Dr. Mustafa Kamal, <i>University of Central Missouri</i>	Katherine Salch, <i>Illinois Central College</i>
John Bundy, <i>DeVry University Chicago</i>	Gary J. Koehler, <i>University of Florida</i>	Javad Shakib, <i>DeVry University</i>
Michael Carney, <i>Finger Lakes Community College</i>	Ronald Krawitz, <i>DeVry University</i>	Charlie Shu, <i>Franklin University</i>
Christopher Cassa, <i>Massachusetts Institute of Technology</i>	Norm Krumpe, <i>Miami University Ohio</i>	Joslyn A. Smith, <i>Florida International University</i>
Dr. Suchindran S. Chatterjee, <i>Arizona State University</i>	Jim Leone, <i>Rochester Institute of Technology</i>	Robert Strader, <i>Stephen F. Austin State University</i>
Tina Comston, <i>Franklin University</i>	Kevin Lillis, <i>St. Ambrose University</i>	Jonathan S. Weissman, <i>Finger Lakes Community College</i>
Lennie Cooper, <i>Miami Dade College</i>	Darren Lim, <i>Siena College</i>	Katherine H. Winters, <i>University of Tennessee Chattanooga</i>
Sherif Elfayoumy, <i>University of North Florida</i>	Hong Lin, <i>DeVry University</i>	Tom Wulf, <i>University of Cincinnati</i>
Henry A Etlinger, <i>Rochester Institute of Technology</i>	Kuber Maharjan, <i>Purdue University College of Technology at Columbus</i>	Qi Yu, <i>Rochester Institute of Technology</i>
	Patricia McDermott-Wells, <i>Florida International University</i>	

Every new edition builds on the suggestions and experiences of prior reviewers and users. I am grateful for the invaluable contributions these individuals have made:

- Tim Andersen, *Boise State University*
 Ivan Bajic, *San Diego State University*
 Ted Bangay, *Sheridan Institute of Technology*
 Ian Barland, *Radford University*
 George Basham, *Franklin University*
 Sambit Bhattacharya, *Fayetteville State University*
 Rick Birney, *Arizona State University*
 Paul Bladek, *Edmonds Community College*
 Joseph Bowbeer, *Vizrea Corporation*
 Timothy A. Budd, *Oregon State University*
 Robert P. Burton, *Brigham Young University*
 Frank Butt, *IBM*
 Jerry Cain, *Stanford University*
 Adam Cannon, *Columbia University*
 Nancy Chase, *Gonzaga University*
 Archana Chidanandan, *Rose-Hulman Institute of Technology*
 Vincent Cicirello, *The Richard Stockton College of New Jersey*
 Teresa Cole, *Boise State University*
 Deborah Coleman, *Rochester Institute of Technology*
 Jose Cordova, *University of Louisiana, Monroe*
 Valentino Crespi, *California State University, Los Angeles*
 Jim Cross, *Auburn University*
 Russell Deaton, *University of Arkansas*
 Geoffrey Decker, *Northern Illinois University*
 H. E. Dunsmore, *Purdue University*
 Robert Duvall, *Duke University*
 Eman El-Sheikh, *University of West Florida*
 John Fendrich, *Bradley University*
 David Freer, *Miami Dade College*
 John Fulton, *Franklin University*
 David Geary, *Sabreware, Inc.*
 Margaret Geroch, *Wheeling Jesuit University*
 Ahmad Ghafarian, *North Georgia College & State University*
 Rick Giles, *Acadia University*
 Stacey Grasso, *College of San Mateo*
 Jianchao Han, *California State University, Dominguez Hills*
 Lisa Hansen, *Western New England College*
 Elliotte Harold
 Eileen Head, *Binghamton University*
 Cecily Heiner, *University of Utah*
 Brian Howard, *Depauw University*
 Lubomir Ivanov, *Iona College*
 Norman Jacobson, *University of California, Irvine*
 Curt Jones, *Bloomsburg University*
 Aaron Keen, *California Polytechnic State University, San Luis Obispo*
 Mugdha Khaladkar, *New Jersey Institute of Technology*
 Elliot Koffman, *Temple University*
 Kathy Liszka, *University of Akron*
 Hunter Lloyd, *Montana State University*
 Youmin Lu, *Bloomsburg University*
 John S. Mallozzi, *Iona College*
 John Martin, *North Dakota State University*
 Jeanna Matthews, *Clarkson University*
 Scott McElfresh, *Carnegie Mellon University*
 Joan McGrory, *Christian Brothers University*
 Carolyn Miller, *North Carolina State University*
 Sandeep R. Mitra, *State University of New York, Brockport*
 Teng Moh, *San Jose State University*
 John Moore, *The Citadel*
 Jose-Arturo Mora-Soto, Jessica Rivero-Espinosa, and Julio-Angel Cano-Romero, *University of Madrid*
 Faye Navabi, *Arizona State University*
 Parviz Partow-Navid, *California State University, Los Angeles*
 Kevin O’Gorman, *California Polytechnic State University, San Luis Obispo*
 Michael Olan, *Richard Stockton College*
 Kevin Parker, *Idaho State University*
 Jim Perry, *Ulster County Community College*
 Cornel Pokorny, *California Polytechnic State University, San Luis Obispo*
 Roger Priebe, *University of Texas, Austin*
 C. Robert Putnam, *California State University, Northridge*
 Kai Qian, *Southern Polytechnic State University*
 Cyndi Rader, *Colorado School of Mines*
 Neil Rankin, *Worcester Polytechnic Institute*
 Brad Rippe, *Fullerton College*
 Pedro I. Rivera Vega, *University of Puerto Rico, Mayaguez*
 Daniel Rogers, *SUNY Brockport*
 Chaman Lal Sabharwal, *Missouri University of Science and Technology*
 John Santore, *Bridgewater State College*
 Carolyn Schauble, *Colorado State University*
 Brent Seales, *University of Kentucky*
 Christian Shin, *SUNY Geneseo*
 Jeffrey Six, *University of Delaware*
 Don Slater, *Carnegie Mellon University*
 Ken Slonneger, *University of Iowa*
 Donald Smith, *Columbia College*
 Stephanie Smullen, *University of Tennessee, Chattanooga*
 Monica Sweat, *Georgia Institute of Technology*
 Peter Stanchev, *Kettering University*
 Shannon Tauro, *University of California, Irvine*
 Ron Taylor, *Wright State University*
 Russell Tessier, *University of Massachusetts, Amherst*
 Jonathan L. Tolsted, *North Dakota State University*
 David Vineyard, *Kettering University*
 Joseph Vybihal, *McGill University*
 Xiaoming Wei, *Iona College*
 Todd Whittaker, *Franklin University*
 Robert Willhoft, *Roberts Wesleyan College*
 Lea Wittie, *Bucknell University*
 David Womack, *University of Texas at San Antonio*
 David Woolbright, *Columbus State University*
 Catherine Wyman, *DeVry University*
 Arthur Yanushka, *Christian Brothers University*
 Salih Yurttas, *Texas A&M University*

CONTENTS

PREFACE iii

CHAPTER 1 INTRODUCTION 1

- 1.1 Computer Programs 2
- 1.2 The Anatomy of a Computer 3
- 1.3 The Java Programming Language 6
- 1.4 Becoming Familiar with Your Programming Environment 8
- 1.5 Analyzing Your First Program 12
- 1.6 Errors 15
- 1.7 Problem Solving: Algorithm Design 16

CHAPTER 2 INTRODUCTION TO OBJECTS AND CLASSES 33

- 2.1 Objects and Classes 34
- 2.2 Variables 36
- 2.3 Working with Objects 43
- 2.4 The API Documentation 47
- 2.5 Instance Variables and Encapsulation 50
- 2.6 Specifying the Public Interface of a Class 54
- 2.7 Providing the Class Implementation 61
- 2.8 Unit Testing 66
- 2.9 Object References 71
- 2.10 Graphical Applications 76

CHAPTER 3 FUNDAMENTAL DATA TYPES 107

- 3.1 Numbers 108
- 3.2 Arithmetic 115
- 3.3 Input and Output 123
- 3.4 Problem Solving: First Do it By Hand 130
- 3.5 Strings 132

CHAPTER 4 DECISIONS 155

- 4.1 The if Statement 156
- 4.2 Comparing Values 162
- 4.3 Multiple Alternatives 172

- 4.4 Nested Branches **176**
- 4.5 Problem Solving: Flowcharts **183**
- 4.6 Problem Solving: Selecting Test Cases **186**
- 4.7 Boolean Variables and Operators **189**
- 4.8 Application: Input Validation **194**

CHAPTER 5 LOOPS **217**

- 5.1 The while Loop **218**
- 5.2 Problem Solving: Hand-Tracing **225**
- 5.3 The for Loop **230**
- 5.4 The do Loop **238**
- 5.5 Application: Processing Sentinel Values **239**
- 5.6 Problem Solving: Storyboards **245**
- 5.7 Common Loop Algorithms **248**
- 5.8 Nested Loops **255**
- 5.9 Application: Random Numbers and Simulations **259**
- 5.10 Using a Debugger **264**

CHAPTER 6 ARRAYS AND ARRAY LISTS **289**

- 6.1 Arrays **290**
- 6.2 Array Lists **299**
- 6.3 The Enhanced for Loop **304**
- 6.4 Array Algorithms **306**
- 6.5 Problem Solving: Adapting Algorithms **317**
- 6.6 Problem Solving: Discovering Algorithms by Manipulating Physical Objects **322**
- 6.7 Two-Dimensional Arrays **326**
- 6.8 Regression Testing **333**

CHAPTER 7 DESIGNING CLASSES **357**

- 7.1 Discovering Classes **358**
- 7.2 Designing Good Methods **359**
- 7.3 Preconditions and Postconditions **368**
- 7.4 Problem Solving: Patterns for Object Data **372**
- 7.5 Static Variables and Methods **377**
- 7.6 Packages **382**
- 7.7 Unit Test Frameworks **389**

CHAPTER 8 INTERFACES AND POLYMORPHISM **405**

- 8.1 Using Interfaces for Algorithm Reuse **406**
- 8.2 Converting Between Class and Interface Types **412**
- 8.3 Polymorphism **414**
- 8.4 The Comparable Interface **415**
- 8.5 Using Interfaces for Callbacks **419**
- 8.6 Inner Classes **423**
- 8.7 **Mock Objects** **425**
- 8.8 Event Handling **426**
- 8.9 Building Applications with Buttons **432**
- 8.10 Processing Timer Events **436**
- 8.11 Mouse Events **439**

CHAPTER 9 INHERITANCE **455**

- 9.1 Inheritance Hierarchies **456**
- 9.2 Implementing Subclasses **460**
- 9.3 Overriding Methods **465**
- 9.4 Object: The Cosmic Superclass **479**

CHAPTER 10 INPUT/OUTPUT AND EXCEPTION HANDLING **495**

- 10.1 Reading and Writing Text Files **496**
- 10.2 Text Input and Output **501**
- 10.3 Command Line Arguments **509**
- 10.4 Exception Handling **516**
- 10.5 Application: Handling Input Errors **526**

CHAPTER 11 OBJECT-ORIENTED DESIGN **541**

- 11.1 Classes and Their Responsibilities **542**
- 11.2 Relationships Between Classes **545**
- 11.3 Application: Printing an Invoice **551**

CHAPTER 12 RECURSION **569**

- 12.1 Triangle Numbers **570**
- 12.2 Recursive Helper Methods **578**
- 12.3 The Efficiency of Recursion **580**
- 12.4 Permutations **585**

- 12.5 Mutual Recursion **590**
- 12.6 Backtracking **596**

CHAPTER 13 SORTING AND SEARCHING **611**

- 13.1 Selection Sort **612**
- 13.2 Profiling the Selection Sort Algorithm **615**
- 13.3 Analyzing the Performance of the Selection Sort Algorithm **618**
- 13.4 Merge Sort **623**
- 13.5 Analyzing the Merge Sort Algorithm **626**
- 13.6 Searching **630**
- 13.7 Problem Solving: Estimating the Running Time of an Algorithm **635**
- 13.8 Sorting and Searching in the Java Library **640**

CHAPTER 14 THE JAVA COLLECTIONS FRAMEWORK **653**

- 14.1 An Overview of the Collections Framework **654**
- 14.2 Linked Lists **657**
- 14.3 Sets **663**
- 14.4 Maps **668**
- 14.5 Stacks, Queues, and Priority Queues **674**
- 14.6 Stack and Queue Applications **677**

CHAPTER 15 BASIC DATA STRUCTURES **697**

- 15.1 Implementing Linked Lists **698**
- 15.2 Implementing Array Lists **713**
- 15.3 Implementing Stacks and Queues **717**
- 15.4 Implementing a Hash Table **723**

CHAPTER 16 TREE STRUCTURES **743**

- 16.1 Basic Tree Concepts **744**
- 16.2 Binary Trees **748**
- 16.3 Binary Search Trees **753**
- 16.4 Tree Traversal **762**
- 16.5 Red-Black Trees **768**
- 16.6 Heaps **775**
- 16.7 The Heapsort Algorithm **786**

CHAPTER 17 GENERIC CLASSES **801**

- 17.1 Generic Classes and Type Parameters **802**
- 17.2 Implementing Generic Types **803**
- 17.3 Generic Methods **807**
- 17.4 Constraining Type Parameters **809**
- 17.5 Type Erasure **813**

CHAPTER 18 GRAPHICAL USER INTERFACES **823**

- 18.1 Layout Management **824**
- 18.2 Processing Text Input **828**
- 18.3 Choices **834**
- 18.4 Menus **845**
- 18.5 Exploring the Swing Documentation **851**

CHAPTER 19 STREAMS AND BINARY INPUT/OUTPUT **863**

- 19.1 Readers, Writers, and Streams **864**
- 19.2 Binary Input and Output **865**
- 19.3 Random Access **869**
- 19.4 Object Streams **875**

CHAPTER 20 MULTITHREADING (WEB ONLY) 

- 20.1 Running Threads
- 20.2 Terminating Threads
- 20.3 Race Conditions
- 20.4 Synchronizing Object Access
- 20.5 Avoiding Deadlocks
- 20.6 Application: Algorithm Animation

CHAPTER 21 INTERNET NETWORKING (WEB ONLY) 

- 21.1 The Internet Protocol
- 21.2 Application Level Protocols
- 21.3 A Client Program
- 21.4 A Server Program
- 21.5 URL Connections

CHAPTER 22 RELATIONAL DATABASES (WEB ONLY) 

- 22.1 Organizing Database Information
- 22.2 Queries
- 22.3 Installing a Database
- 22.4 Database Programming in Java
- 22.5 Application: Entering an Invoice

CHAPTER 23 XML (WEB ONLY) 

- 23.1 XML Tags and Documents
- 23.2 Parsing XML Documents
- 23.3 Creating XML Documents
- 23.4 Validating XML Documents

CHAPTER 24 WEB APPLICATIONS (WEB ONLY) 

- 24.1 The Architecture of a Web Application
- 24.2 The Architecture of a JSF Application
- 24.3 JavaBeans Components
- 24.4 Navigation Between Pages
- 24.5 JSF Components
- 24.6 A Three-Tier Application

APPENDICES

- APPENDIX A** THE BASIC LATIN AND LATIN-1 SUBSETS OF UNICODE **A-1**
 - APPENDIX B** JAVA OPERATOR SUMMARY **A-5**
 - APPENDIX C** JAVA RESERVED WORD SUMMARY **A-7**
 - APPENDIX D** THE JAVA LIBRARY **A-9**
 - APPENDIX E** JAVA SYNTAX SUMMARY **A-53**
 - APPENDIX F** TOOL SUMMARY **A-64**
 - APPENDIX G** NUMBER SYSTEMS **A-68**
 - APPENDIX H** UML SUMMARY **A-76**
 - APPENDIX I** JAVA LANGUAGE CODING GUIDELINES **A-79**
 - APPENDIX J** HTML SUMMARY **A-86**
- GLOSSARY **G-1**
- INDEX **I-1**
- CREDITS **C-1**

ALPHABETICAL LIST OF SYNTAX BOXES

Arrays	291
Array Lists	299
Assertion	369
Assignment	41
Calling a Superclass Method	465
Cast	119
Catching Exceptions	518
Class Declaration	57
Comparisons	163
Constant Declaration	112
Constructor with Superclass Initializer	471
Declaring a Generic Class	804
Declaring a Generic Method	808
Declaring an Interface	407
for Statement	230
if Statement	158
Implementing an Interface	409
Importing a Class from a Package	49
Input Statement	123
Instance Variable Declaration	51
Java Program	13
Object Construction	44
Package Specification	384
Subclass Declaration	462
The Enhanced for Loop	305
The finally Clause	522
The instanceof Operator	483
The throws Clause	521
Throwing an Exception	516
Two-Dimensional Array Declaration	327
while Statement	219
Variable Declaration	37