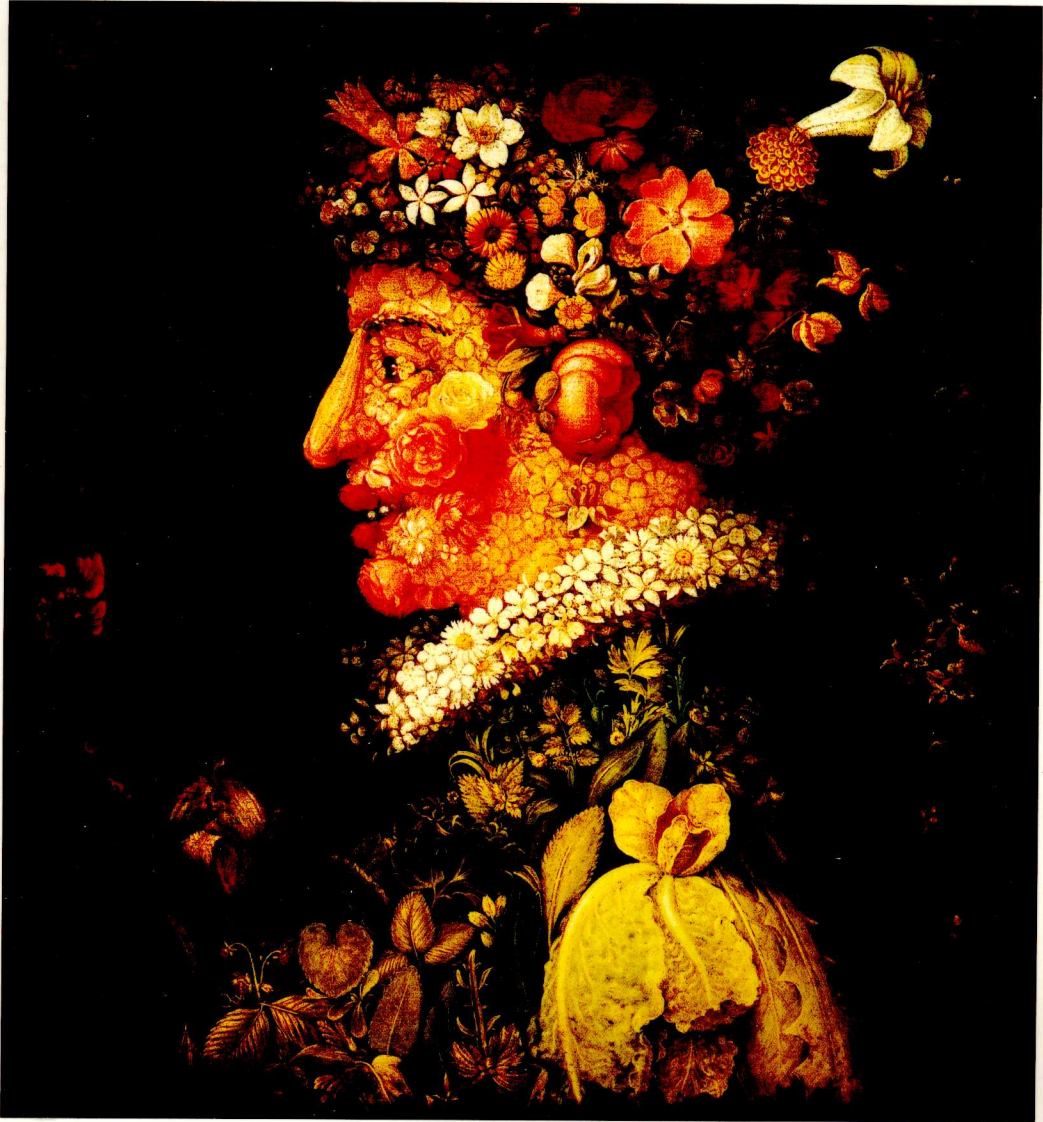


SECOND EDITION

# programming.java™

An Introduction to Programming Using Java



**DECKER**

**HIRSHFIELD**



986127

# programming.java

## **An Introduction to Programming Using Java™**

*Second Edition*

**Rick Decker  
Stuart Hirshfield**



*Hamilton College*



**Brooks/Cole**  
Thomson Learning™

Australia • Canada • Mexico • Singapore • Spain • United Kingdom • United States

Sponsoring Editor: *Kallie Swanson*  
Marketing Team: *Nathan Wilbur,*  
*Christina De Veto, Samantha Cabaluna*  
Editorial Assistant: *Grace Fujimoto*  
Production Coordinator: *Kelsey McGee*  
Production Service: *Publishers' Design and*  
*Production Services, Inc.*  
Manuscript Editor: *Connie Day*

Permissions Editor: *Mary Kay Hancharick*  
Cover Design: *Laurie Albrecht*  
Typesetting: *Publishers' Design and Production*  
*Services, Inc.*  
Cover Printing: *Phoenix Color Corporation*  
Printing and Binding: *R. R. Donnelley and Sons/*  
*Crawfordsville*

COPYRIGHT © 2000 by Brooks/Cole  
A division of Thomson Learning  
The Thomson Learning logo is a trademark used herein under license.

*For more information, contact:*  
BROOKS/COLE  
511 Forest Lodge Road  
Pacific Grove, CA 93950 USA  
[www.brookscole.com](http://www.brookscole.com)

All rights reserved. No part of this work may be reproduced, transcribed or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, Web distribution, or information storage and/or retrieval systems—without the prior written permission of the publisher.

*For permission to use material from this work, contact us by*  
Web: [www.thomsonrights.com](http://www.thomsonrights.com)  
fax: 1-800-730-2215  
phone: 1-800-730-2214

Printed in United States of America

10 9 8 7 6 5 4 3 2

#### **Library of Congress Cataloging-in-Publication Data**

Decker, Rick.

Programming Java : an introduction to programming using Java /  
Rick Decker, Stuart Hirshfield.—2nd ed.

p. cm.

ISBN 0-534-37109-4 (pbk.)

I. Java (Computer program language) I. Hirshfield, Stuart.

II. Title.

QA76.73.J38D44 1999

005.13'3—dc21

99-38991



To our students,  
understanding as we do the ups,

*Many are our joys  
In youth, but oh! What happiness to live  
When every hour brings palpable access  
Of knowledge, when all knowledge is delight  
And sorrow is not there!*

—William Wordsworth, *The Prelude*

And the downs,

*My life is one demd horrid grind.*

—Charles Dickens, *Oliver Twist*

---

# PREFACE

This second edition of “programming dot java” (like the first) is intended to be used in a one- or two-semester introduction to programming using the Java™ language. We assume no prior programming experience on the part of our audience. In the tradition of new releases of software, we’ll begin with a revision history of this product, describing the features of the original version (“edition,” if you prefer), and the changes we have made.

## Revision History

### programming java 1.0

**Hands-on experience.** In a separate lab manual—available in print and on-line formats—we provide students with detailed, directed, experimental exercises which help them to explore firsthand the principles of OOP and Java in a controlled fashion. These exercises are integrated precisely with the textual material and bring the static text material to life. The text introduces students to new concepts. As they read the text, students encounter a number of sample programs that illustrate the new concepts. Then, in the laboratory, they experiment with “Lablet” programs and extend them using what they have learned from the text. All the sample programs will be provided on a disk that will accompany the lab manual.

**Objects from the start.** It’s almost impossible to avoid classes, objects, and inheritance in a Java program. This means that we could build in an object-oriented approach from the very beginning, rather than just tacking it on later. Specifically, this text introduces students to Java and OOP from the top down. That is, we start by presenting the empowering features of Java and OOP—classes, packages, and inheritance—and defer the algorithmic details until later. After a quick generic introduction to computers and programming, roughly the first third of the text is devoted to using and experimenting with Java’s Abstract Windowing Toolkit (the “AWT”). Doing so allows us to provide students with lots of algorithmically simple code to read and to use, all of which emphasizes basic OOP notations and ideas while providing tangible graphical output. Our overriding motivation in this approach is to make sure our students are introduced to object-oriented design early, before any bad habits can set in. By the end of this section of the course, students will be able to use the AWT to describe arbitrarily complex graphical user interfaces (GUIs). More important, they will be completely conversant in the basics of OOP and the use of Java packages.

**Graphical orientation.** Java contains a rich class library containing complete, platform-independent support for event-driven, graphically-oriented programming.

That is, users can interact with Java programs in a modeless manner by means of a contemporary graphical user interface, rather than by responding to program prompts by typing text into a console window. Just as the language dictates a reliance on object-oriented design, the class library makes it virtually impossible to avoid an approach that emphasizes code reuse. From the instructor's point of view, we gain a natural context for discussing important principles of software design, and our students can build visually interesting programs from the very start. After all, our audience has been raised on programs that have a rich graphical user interface: Java allows them to build the kind of programs they're used to seeing.

### **programming java 2.0**

**Java 2 compatibility.** When we were writing version 1.0 of this text, the Java language was in the process of a major change, placing us in the uncomfortable position of releasing a book that was aimed at a moving target. Realizing that it would be a while before compiler writers caught up with the language change, we gritted our teeth and wrote it so that it discussed both the old and the (then) new versions of Java. Just to keep us on our toes (or so it seemed), as we were writing the current version, Sun Microsystems introduced yet another version—Java 2. Fortunately, this change entailed only the smallest of modifications, other than a decision to omit—for the moment—any detailed coverage of the new library packages (particularly Swing) that comprise Java 2. This means that the labs, lablets, and language features covered here are compatible with any version of Java with number 1.1 or higher.

**New Lablets, new labs.** We've streamlined many of the sample Lablets and completely replaced some, in the interest of not clobbering our readers with too much detail. The new model of event handling introduced in Java 1.1 necessitated changes in most of the Lablets, and, most important, we squashed a number of bugs that cropped up in the earlier programs. Of course, the changes in the Lablets resulted in a number of changes in the associated labs, as well.

**New features.** This version of the text includes review self-test questions at the end of nearly every section, with answers provided near the end of each chapter. We've also added end-of chapter reference sections that describe in compact form all the classes introduced in the chapter. In the back of the book, we've included a glossary of new terms and a complete rewrite of the atrocious index that came with the first version.

## **Coverage**

You won't find the traditional CS 1 chapters, "Loops" and "Selection Statements," here. Our approach, as we've explained, is to cover object-oriented design and programming from the beginning, and to adopt a "just in time" approach to the algorithmic language features. **Chapter 1** provides a background context, discussing the

evolution of programming, the Internet and the Worldwide Web, and providing an overview of Java.

The next three chapters introduce the AWT fundamentals. **Chapter 2** discusses the Applet class and graphical programming. **Chapter 3** introduces the basic graphical user interface components, such as buttons, labels, textfields, and checkboxes. The Lablet for this chapter uses all the components in the front end of an online ordering program. **Chapter 4** continues the investigation of the AWT classes by discussing containers, layouts, windows, frames, dialogs, and menus.

**Chapter 5** begins our transition from using the AWT classes for visual design to writing programs that actively interact with the user. Here we introduce the Java language features that students will need, discussing primitive types, identifiers, scope, access, expressions, and statements. **Chapter 6** introduces event-driven programming and, now that they're necessary, the Java selection statements. **Chapter 7** recapitulates everything that we've introduced and then explores the design process in detail, moving from a vaguely-worded description of a problem to a fully functional simulation of an automatic teller machine. **Chapter 8** discusses arrays and the String class, introducing loops. We begin to discuss algorithmic programming by producing a visual sorting demonstration.

The remaining four chapters cover the other classes needed to produce all but the more arcane Java programs. **Chapter 9** introduces exceptions. **Chapter 10** discusses file input and output, culminating with the design and exploration of a word processor. **Chapter 11** introduces threads. **Chapter 12** concludes the book, describing how to make an applet interact with the Web, using sounds, images, and animation.

*programming.java* contains more material than most instructors will wish to introduce in a single semester. Many people we've talked to indicate that they cover most of Chapters 1 through 8 and then, depending on their interests and the interests of their students pick a topic or two from the remaining four chapters. We've heard that some schools have what might be called a "CS 1.5" course—covering Chapters 8 through 12 along with a healthy dose of software engineering—either as a requirement or an elective between CS 1 and the traditional Data Structures course that usually follows CS 1.

## Thanks

*programming.java*, *second edition* wouldn't exist were it not for the efforts of a number of diligent and helpful people, many of whom we're proud to count among our friends. Our heartfelt thanks go out to the people at Brooks/Cole, especially Kallie Swanson, our editor; Grace Fujimoto, Brooks/Cole's editorial assistant for computer science titles; and Kelsey McGee, the production coordinator; Nathan Wilbur, senior marketing manager for engineering and computer science; and Mark Bergeron, of Publishers' Design and Production Services, Inc. Thanks also go to Connie Day, for reading the manuscript and magically transforming our sow's ear into a silk purse of accepted standards of punctuation, grammar, and

orthography, and to Michael Fry of Lebanon Valley College, Kent Jones of Whitworth College, Robert Kline of West Chester University, J. Richard Rinewalt of Texas Christian University, Celia Schahczenski of Montana Tech of the University of Montana, and Susanne Steiger of Massachusetts Bay Community College for their detailed, thorough, and enormously helpful reviews of this edition. Special thanks, finally, go to our families for putting up with frequently absent and always bemused husbands and fathers. As Brian Dennehy said about his family in his Tony acceptance speech, “they paid for this with that most precious of coins, time.” Writing and producing a book is a process that ranks right up there on the misery scale with cholera, except that it takes longer. It can never be called a pleasant experience, but the help, support, and warmth of all involved at least made it bearable.

*Rick Decker*

*Stuart Hirshfield*



---

---

# CONTENTS

## Chapter One BACKGROUND

I

<b>1.1</b>	<b>The Evolution of Programming</b>	<b>1</b>
	The Not-So-Good Old Days	2
	Help Arrives	5
	Review Questions	7
<b>1.2</b>	<b>The Internet and the Worldwide Web</b>	<b>7</b>
	The Worldwide Web	8
	HTML	10
	Review Questions	12
<b>1.3</b>	<b>Along Comes Java</b>	<b>12</b>
	Smart Toasters	12
	Java, Meet Web; Web, Meet Java	13
	Applications and Applets	14
	Review Questions	17
<b>1.4</b>	<b>An Overview of Java</b>	<b>18</b>
	Syntax	18
	Language Features	19
	Objects and Classes	20
	Inheritance	22
	Libraries	23
	Review Questions	24
<b>1.5</b>	<b>Hands On</b>	<b>24</b>
<b>1.6</b>	<b>Resources Online</b>	<b>27</b>
<b>1.7</b>	<b>Summary</b>	<b>27</b>
<b>1.8</b>	<b>Exercises</b>	<b>28</b>
<b>1.9</b>	<b>Answers to Review Questions</b>	<b>30</b>

## Chapter Two APPLET

33

<b>2.1</b>	<b>The Applet Class</b>	<b>33</b>
	Learning From a Simple Applet	33
	Review Questions	36

<b>2.2</b>	<b>Methods, Inheritance, and Overriding</b>	<b>36</b>
	Java Methods	37
	Inheritance and Overriding	38
	Review Questions	41
<b>2.3</b>	<b>Graphical Programming</b>	<b>41</b>
	The Graphics Class	42
	Using the Graphics Class (and Learning Some Programming in the Process)	43
	The Color Class	46
	The Font Class	47
	Positions and Sizes: The Classes Point, Dimension, and Rectangle	47
	Review Questions	50
<b>2.4</b>	<b>Hands On</b>	<b>50</b>
	Pass 1: Painting	50
	Pass 2: Clarifying by Complicating	54
	Pass 3: Methods of Our Own	56
	Review Questions	59
<b>2.5</b>	<b>Summary</b>	<b>59</b>
<b>2.6</b>	<b>Exercises</b>	<b>60</b>
<b>2.7</b>	<b>Answers to Review Questions</b>	<b>64</b>
<b>2.8</b>	<b>Class References</b>	<b>65</b>
	java.applet.Applet	65
	java.awt.Color	66
	java.awt.Dimension	66
	java.awt.Font	67
	java.awt.Graphics	67
	java.awt.Point	69
	java.awt.Rectangle	70

## Chapter Three WIDGETS

73

<b>3.1</b>	<b>Components</b>	<b>73</b>
	Component's Graphical Methods	74
	Review Questions	76
<b>3.2</b>	<b>Textual Widgets</b>	<b>76</b>
	The Label Class	76
	The TextComponent Class	77
	The TextField Class	78

Using Textfields	79
The TextArea Class	81
Using TextAreas	83
Review Questions	84
<b>3.3 Active Widgets</b>	<b>85</b>
The Button Class	85
The Checkbox Class	86
The CheckboxGroup Class	87
The Choice Class	88
The List Class	89
Review Questions	91
<b>3.4 Hands On</b>	<b>91</b>
<b>3.5 Summary</b>	<b>95</b>
<b>3.6 Exercises</b>	<b>96</b>
<b>3.7 Answers to Review Questions</b>	<b>99</b>
<b>3.8 Class References</b>	<b>100</b>
java.awt.Button	100
java.awt.Checkbox	100
java.awt.CheckboxGroup	101
java.awt.Choice	101
java.awt.Component	101
java.awt.Cursor	104
java.awt.Label	104
java.awt.List	105
java.awt.TextArea	107
java.awt.TextComponent	107
java.awt.TextField	108

## Chapter Four VISUAL DESIGN

109

<b>4.1 Containers</b>	<b>109</b>
Container Organization Methods	110
The Containment Hierarchy	112
The Panel Class	113
Review Questions	113
<b>4.2 Layouts</b>	<b>113</b>
Container Layout Methods	114
The FlowLayout Class	114
The BorderLayout Class	116
The GridLayout Class	119

No Layout at All	121	
Review Questions	122	
<b>4.3 Other Containers, Other Details</b>		<b>122</b>
The Canvas Class	123	
Windows	125	
Frames	126	
Dialogs	128	
Review Questions	134	
<b>4.4 Menus</b>	<b>135</b>	
MenuComponents	136	
MenuBar	137	
Menus	137	
MenuItem	138	
CheckboxMenuItem	138	
PopupMenu	139	
A Menu Sample	139	
Review Questions	141	
<b>4.5 Hands On</b>	<b>141</b>	
Designing the Lablet	141	
The Lablet Code	143	
<b>4.6 Summary</b>	<b>146</b>	
<b>4.7 Exercises</b>	<b>147</b>	
<b>4.8 Answers to Review Questions</b>		<b>152</b>
<b>4.9 Class References</b>	<b>152</b>	
java.awt.BorderLayout	152	
java.awt.Canvas	153	
java.awt.CardLayout	153	
java.awt.CheckboxMenuItem	154	
java.awt.Container	155	
java.awt.Dialog	156	
java.awt.FlowLayout	156	
java.awt.Frame	157	
java.awt.GridLayout	157	
java.awt.Menu	158	
java.awt.MenuBar	159	
java.awt.MenuComponent	159	
java.awt.MenuItem	160	
java.awt.MenuShortcut	160	
java.awt.Panel	161	
java.awt.PopupMenu	161	
java.awt.Window	162	

**Chapter Five JAVA LANGUAGE BASICS****163**

<b>5.1</b>	<b>Primitive Types</b>	<b>163</b>
	Integers	164
	Floating-Point Numbers	165
	Characters	166
	The boolean Type	166
	Review Questions	166
<b>5.2</b>	<b>Identifiers, Keywords, and Variables</b>	<b>166</b>
	Variables	167
	Scope	169
	The Modifiers <code>static</code> and <code>final</code>	171
	Brief Interlude: Packages	173
	Access Modifiers	174
	<code>private</code> Access	174
	Package Access	176
	<code>public</code> Access	177
	Review Questions	178
<b>5.3</b>	<b>Operators and Expressions</b>	<b>178</b>
	Numeric Operators	178
	The <code>Math</code> Class	182
	Bitwise Operators (optional)	184
	Boolean Operators	187
	Complicated boolean Expressions	190
	Review Questions	194
<b>5.4</b>	<b>Assignments and Statements</b>	<b>194</b>
	Assignments	194
	Class Variables versus Primitive Variables	196
	Miscellaneous Operators	199
	Statements	199
	Review Questions	201
<b>5.5</b>	<b>Hands On</b>	<b>201</b>
	Designing the Lablet, I	202
	Designing the Lablet, II	203
	The Lablet Code	204
<b>5.6</b>	<b>Summary</b>	<b>209</b>
<b>5.7</b>	<b>Exercises</b>	<b>212</b>
<b>5.8</b>	<b>Answers to Review Questions</b>	<b>218</b>
<b>5.9</b>	<b>Class References</b>	<b>218</b>
	<code>java.lang.Math</code>	218

java.util.Random	219
java.math.BigInteger	220

## Chapter Six EVENTS AND ACTIONS

223

<b>6.1 More Java Programming</b>	<b>223</b>
The if Statement	223
Common Problems with if	227
The switch Statement	231
Abstract Classes and Interfaces	233
Review Questions	237
<b>6.2 Event-Driven Programming</b>	<b>237</b>
The Delegation Model	238
Review Questions	241
<b>6.3 The AWTEvent Hierarchy</b>	<b>242</b>
Upper-Level Event Classes	242
Action Events	243
Adjustment Events and Scrollbars	245
Input Events	245
Item Events	246
Key Events	247
Mouse Events	248
Text Events	248
Review Questions	249
<b>6.4 Listeners</b>	<b>249</b>
Listener Interfaces	249
ActionListener	250
ItemListener	250
KeyListener	250
MouseListener	251
MouseMotionListener	251
TextListener	251
Adapters	251
Review Questions	252
<b>6.5 Hands On</b>	<b>252</b>
The GalaEvents Lablet	252
The SketchPad Lablet (optional)	256
Review Question	264
<b>6.6 Summary</b>	<b>264</b>
<b>6.7 Exercises</b>	<b>267</b>

<b>6.8</b>	<b>Answers to Review Questions</b>	<b>271</b>
<b>6.9</b>	<b>Class References</b>	<b>272</b>
	java.awt.event.ActionEvent	272
	java.awt.event.ActionListener	272
	java.awt.event.AdjustmentEvent	272
	java.awt.event.AdjustmentListener	273
	java.awt.AWTEvent	273
	java.awt.event.ComponentEvent	273
	java.util.EventObject	273
	java.awt.event.FocusAdapter	274
	java.awt.event.FocusEvent	274
	java.awt.event.FocusListener	274
	java.awt.event.InputEvent	275
	java.awt.event.ItemEvent	275
	java.awt.event.ItemListener	275
	java.awt.event.KeyAdapter	276
	java.awt.event.KeyEvent	276
	java.awt.event.KeyListener	277
	java.awt.event.MouseAdapter	277
	java.awt.event.MouseMotionAdapter	277
	java.awt.event.MouseEvent	277
	java.awt.event.MouseListener	278
	java.awt.event.MouseMotionListener	278
	java.awt.Scrollbar	278
	java.awt.event.TextEvent	279
	java.awt.event.TextListener	279
	java.awt.event.WindowAdapter	279
	java.awt.event.WindowEvent	280
	java.awt.event.WindowListener	280

## **Chapter Seven    METHODICAL PROGRAMMING    281**

<b>7.1</b>	<b>Method Recap</b>	<b>281</b>
	Method Signatures	282
	Calling Methods	283
	Arguments	285
	Value Arguments, Reference Arguments	287
	Review Questions	289
<b>7.2</b>	<b>Step 1: Specification</b>	<b>289</b>
	Specification	290
	Review Questions	292

<b>7.3 Step 2: Determine the Classes</b>	<b>292</b>
Layout	293
Filling in the Details	296
Review Questions	298
<b>7.4 Step 3: Determine the Methods</b>	<b>298</b>
Top-Level Decomposition	299
Filling in the Details, Again	301
Review Questions	304
<b>7.5 Step 3, Continued</b>	<b>304</b>
A New Class	306
Cleanup	316
Review Questions	318
<b>7.6 The ATM Applet</b>	<b>318</b>
<b>7.7 Hands On</b>	<b>324</b>
<b>7.8 Summary</b>	<b>325</b>
<b>7.9 Exercises</b>	<b>327</b>
<b>7.10 Answers to Review Questions</b>	<b>331</b>
<b>7.11 Class References</b>	<b>332</b>
java.lang.Boolean	332
java.lang.Character	332
java.lang.Double	333
java.lang.Integer	334
java.lang.Number	334

## Chapter Eight COLLECTIONS

335

<b>8.1 Loops</b>	<b>335</b>
The do Loop	336
The while Loop	337
The for Loop	338
Common Problems with Loops	340
Review Questions	341
<b>8.2 Arrays</b>	<b>342</b>
Declaring Arrays	342
Accessing Array Elements	344
Arrays and Loops	344
Multidimensional Arrays	348
Heterogeneous Arrays	351
Review Questions	354



<b>8.3</b>	<b>Sorting</b>	<b>354</b>	
	Selection Sort	354	
	Insertion Sort	356	
	Quicksort	358	
	Review Questions	361	
<b>8.4</b>	<b>Vectors</b>	<b>362</b>	
	The Vector Class	363	
	Inspectors	364	
	Modifiers	365	
	When to Use Vectors	366	
	Review Questions	367	
<b>8.5</b>	<b>Strings</b>	<b>367</b>	
	The String Class	368	
	Access and Comparison	369	
	Builders	371	
	Using Strings for Conversion	372	
	Review Questions	373	
<b>8.6</b>	<b>Hands On</b>	<b>374</b>	
	Designing the Lablet	374	
	Exploring the Lablet	375	
	Review Questions	381	
<b>8.7</b>	<b>Summary</b>	<b>382</b>	
<b>8.8</b>	<b>Exercises</b>	<b>385</b>	
<b>8.9</b>	<b>Answers to Review Questions</b>	<b>394</b>	
<b>8.10</b>	<b>Class References</b>	<b>395</b>	
	java.lang.String	395	
	java.util.Vector	395	

## Chapter Nine EXCEPTIONS

399

<b>9.1</b>	<b>Exceptions</b>	<b>399</b>	
	Exception Subclasses	400	
	Methods That Throw Exceptions	402	
	Review Questions	405	
<b>9.2</b>	<b>Handling Exceptions</b>	<b>405</b>	
	try and catch	405	
	Exception Propagation	407	
	Throwing Exceptions	408	