Contents

Preface iii

0 Computers, Objects, and Java 1

0.1 Welcome 4
0.2 Why Study Programming? 4
0.3 Why Java? 4
0.4 What Is Object-Oriented Programming? 7
0.5 What Is a Computer? 10
0.6 The Internet and the World Wide Web 12
0.7 Programming Languages 14
Chapter Summary 16
Exercises 17

1 Java Program Development 19

1.1 Introduction 22
1.2 Java Applications and Applets 22
1.3 Compiling and Executing a Java Program 24
1.4 Designing Good Programs 28
Grace Hopper and the First Computer Bug 37
From the Java Library: `System` and `PrintStream` 38
1.5 Qualified Names in Java 41
In the Laboratory: Editing, Compiling, and Running an Applet 42
Chapter Summary 50
Answers to Self-Study Exercises 52
Exercises 52

2 Objects: Defining, Creating, and Using 56

2.1 Introduction 58

2.2 Interacting Objects 58

2.3 The Java Class Hierarchy—Classifying Objects 60

2.4 Class Definition 61

2.5 Case Study: Simulating a CyberPet 66

Object-Oriented Design: Basic Principles 89

Alan Kay and the Smalltalk Language 89

Java Language Summary 90

From the Java Library: BufferedReader, String, Integer 100

In the Laboratory: The Circle Class 104

Chapter Summary 107

Answers to Self-Study Exercises 109

Exercises 111

3 Methods: Communicating with Objects 115

3.1 Introduction 118

3.2 Passing Information to an Object 118

3.3 Constructor Methods 126

3.4 Retrieving Information from an Object 131

3.5 Passing a Value and Passing a Reference 135

3.6 Flow of Control: Selection Control Structures 137

3.7 The Improved CyberPet 146

From the Java Library: Object 149

Intelligent Agents 149

Object-Oriented Design: Class Inheritance 152

In the Laboratory: Feeding CyberPet 152

Java Language Summary 155

Chapter Summary 157

Answers to Self-Study Exercises 158

Exercises 160

4 Applets: Programming for the World Wide Web 166

4.1 Introduction 168
4.2 The Applet Class 168
4.3 Applet Subclasses 173
4.4 A Simple Applet 174
4.5 Event-Driven Programming 183
4.6 Case Study: The CyberPetApplet 190
Tim Berners-Lee, Creator of the WWW 200
Object-Oriented Design: Inheritance and Polymorphism 201
From the Java Library: Image 204
In the Laboratory: CyberPetApplet 206
Java Language Summary 210
Chapter Summary 212
Answers to Self-Study Exercises 213
Exercises 215

5 Java Data and Operators 219

5.1 Introduction 222
5.2 Programming = Representation + Action 222
5.3 Boolean Data and Operators 223
5.4 The Boolean-based CyberPet Model 227
Are We Computers? 229
5.5 Numeric Data and Operators 230
5.6 Case Study: Converting Fahrenheit to Celsius 241
5.7 An Integer-based CyberPet Model 251
Object-Oriented Design: Information Hiding 257
5.8 Character Data and Operators 257
5.9 Example: Character Conversions 262
From the Java Library: The Math Class 265
From the Java Library: The NumberFormat Class 267
5.10 Example: Calculating Compound Interest 268
5.11 Problem Solving = Representation + Action 271

In the Laboratory: The Leap Year Problem 271
Java Language Summary 274
Chapter Summary 277
Answers to Self-Study Exercises 279
Exercises 280

6 Control Structures 284

6.1 Introduction 286
6.2 Flow-of-Control: Repetition Structures 286
6.3 Counting Loops 288
6.4 Example: Car Loan 295
6.5 Conditional Loops 297
6.6 Example: Computing Averages 303
6.7 Example: Data Validation 307
6.8 Case Study: Animated `CyberPet` 311
6.9 Principles of Loop Design 315
Object-Oriented Design: Structured Programming 316
What Can Be Computed? 322
From the Java Library: `TextArea` 324
In the Laboratory: Finding Prime Numbers 327
Cryptography 328
Java Language Summary 333
Chapter Summary 335
Answers to Self-Study Exercises 336
Exercises 340

7 Strings and String Processing 345

7.1 Introduction 348
7.2 `String` Basics 349
7.3 Finding Things within a `String` 354
7.4 Example: Keyword Search 357
From the Java Library: `StringBuffer` 359
7.5 Retrieving Parts of `Strings` 361

7.6 Example: Processing Names and Passwords 363
7.7 Processing Each Character in a String 364
7.8 Case Study: Silly CyberPet String Tricks 366
7.9 Comparing Strings 371
From the Java Library: StringTokenizer 377
Historical Cryptography 379
Object-Oriented Design: The abstract Cipher Class 380
In the Laboratory: Pig Latin Translation 386
Java Language Summary 390
Chapter Summary 390
Answers to Self-Study Exercises 392
Exercises 394

8 Arrays and Array Processing 399

8.1 Introduction 402
8.2 One-Dimensional Arrays 403
8.3 Simple Array Examples 410
8.4 Example: Testing a Die 412
8.5 Case Study: CyberPet Animation 417
Data Compression 420
8.6 Array Algorithms: Sorting 421
8.7 Array Algorithms: Searching 428
8.8 Two-Dimensional Arrays 433
8.9 Multidimensional Arrays 440
Object-Oriented Design: Polymorphic Sorting 443
From the Java Library: Vector 446
8.10 Case Study: Simulating a Card Deck 447
In the Laboratory: A Card-Game Applet 453
Java Language Summary 457
Chapter Summary 457
Answers to Self-Study Exercises 459
Exercises 463

9 Graphical User Interfaces 468

9.1 Introduction 470
From the Java Library: AWT to Swing 470
9.2 The Swing Component Set 474
Object-Oriented Design: Model-View-Controller Architecture 475
9.3 The Java Event Model 478
9.4 Case Study: Designing a Basic GUI 482
9.5 Containers and Layout Managers 495
9.6 Checkboxes, Radio Buttons, and Borders 501
9.7 Menus and Scroll Panes 509
In the Laboratory: The ATM Machine 518
Are Computers Intelligent? 519
Chapter Summary 526
Answers to Self-Study Exercises 528
Exercises 530

10 Graphics and Drawing 533

10.1 Introduction 536
10.2 The Drawing Surface 536
10.3 The `Graphics` Context 536
10.4 The `Color` Class 541
Object-Oriented Design: Reference Constants 548
From the Java Library: `Points` and `Dimensions` 549
10.5 Painting and Drawing Lines and Shapes 551
10.6 Example: The `ShapeDemo` Applet 552
10.7 Graphing Equations 555
10.8 Drawing Bar Charts and Pie Charts 560
10.9 Handling Text in a Graphics Context 567
10.10 Case Study: Interactive Drawing 574
Object-Oriented Design: The Scalable `CyberPet` 579
In the Laboratory: The `SelfPortrait` Class 589
Chapter Summary 591

Answers to Self-Study Exercises 593
Exercises 596

11 Exceptions: When Things Go Wrong 599

11.1 Introduction 602
11.2 Handling Exceptional Conditions 602
11.3 Java's Exception Hierarchy 618
11.4 Error Handling and Robust Program Design 622
11.5 Creating and Throwing Your Own Exceptions 630
From the Java Library: `JOptionPane` 632
In the Laboratory: Measuring Exception Overhead 635
The Y2K Exception? 636
Java Language Summary 642
Chapter Summary 642
Answers to Self-Study Exercises 645
Exercises 648

12 Recursive Problem Solving 651

12.1 Introduction 654
12.2 Recursive Definition 657
12.3 Recursive String Methods 660
12.4 Recursive Array Processing 668
12.5 Example: Drawing (Recursive) Fractals 675
Object-Oriented Design: Tail Recursion 679
Object-Oriented Design: Recursion or Iteration 681
12.6 Flow of Control: The `switch` Multiway Selection Structure 683
Exploring the Mandelbrot Set 683
From the Java Library: `JComboBox` 687
In the Laboratory: The `RecursivePatterns` Applet 692
Java Language Summary 693

Chapter Summary 694
Answers to Self-Study Exercises 695
Exercises 699

13 **Threads and Concurrent Programming** 702

13.1 Introduction 703
13.2 What Is a Thread? 704
From the Java Library: The `Thread` class 708
13.3 Using Threads to Improve Interface Responsiveness 714
13.4 Thread States and Life Cycle 723
13.5 Case Study: Cooperating Threads 725
13.6 Case Study: The Spider and Fly Threads 739
Object-Oriented Design: Inheritance and Polymorphism 752
In the Laboratory: The Spider, the Fly, and the Bee 753
Chapter Summary 757
Answers to Self-Study Exercises 759
Exercises 760

14 **Files, Streams, and Input/Output Techniques** 763

14.1 Introduction 765
14.2 Streams and Files 766
14.3 Case Study: Reading and Writing Text Files 773
14.4 The `File` Class 785
14.5 Example: Reading and Writing Binary Files 789
14.6 Reading and Writing Objects 798
From the Java Library: `JFileChooser` 803
In the Laboratory: The `TextEdit` Program 806
Databases and Personal Privacy 807
Chapter Summary 809
Answers to Self-Study Exercises 811
Exercises 813

15 Sockets and Networking 816

15.1 Introduction 818
15.2 An Overview of Networks 818
15.3 Using Network Resources from an Applet 826
From the Java Library: `java.net.URL` 827
15.4 The Slide Show Applet 829
15.5 Using Network Resources from an Application 834
15.6 Client/Server Communication via `Sockets` 844
15.7 Case Study: Generic Client/Server Classes 849
15.8 Java Network Security Restrictions 863
In the Laboratory: The Internet CyberPet 864
Privacy and the Internet 865
Chapter Summary 870
Answers to Self-Study Exercises 871
Exercises 873

16 Data Structures: Lists, Stacks, and Queues 875

16.1 Introduction 878
16.2 The Linked List Data Structure 878
Object-Oriented Design: The List Abstract Data Type (ADT) 890
16.3 The Stack ADT 895
16.4 The Queue ADT 898
From the Java Library: `java.util.Stack` 900
In the Laboratory: Capital Gains 900
The LISP Language 901
Chapter Summary 906
Answers to Self-Study Exercises 907
Exercises 910

A Coding Conventions 913

Comments 913
Indentation and White Space 915
Naming Conventions 916
Use of Braces 917
File Names and Layout 917
Statements 918
Executable Statements 920
Preconditions and Postconditions 920
Sample Programs 921

B The Java Development Kit 922

The Java Compiler: `javac` 922
The Java Interpreter: `java` 924
The `appletviewer` 925
The Java Archiver `jar` Tool 928
The Java Documentation Tool: `javadoc` 930

C The ASCII and Unicode Character Sets 931

D Java Keywords 932

E Operator Precedence Hierarchy 933

F Advanced Language Features 935

Inner Classes 935
Nested Top-Level versus Member Classes 936
Local and Anonymous Inner Classes 938

G Java Resources 941

Reference Books 941
On-line References 941

Index 943