



Pearson International Edition

Classes and Methods • References • Scanner Class Input • Automatic Boxing
Enhanced for Loop • Interfaces • Inheritance • Polymorphism • Encapsulation
Generics • Streams and File IO • Exception Handling • ArrayList
Linked Lists • Swing GUIs • Threads



Absolute Java

Walter Savitch

Third Edition

Contents

Chapter 1 Getting Started 1

- 1.1 INTRODUCTION TO JAVA 2**
 - Origins of the Java Language ★ 2
 - Objects and Methods 3
 - Applets ★ 4
 - A Sample Java Application Program 5
 - Byte-Code and the Java Virtual Machine 8
 - Class Loader ★ 10
 - Compiling a Java Program or Class 10
 - Running a Java Program 11
 - Tip: Error Messages 12

- 1.2 EXPRESSIONS AND ASSIGNMENT STATEMENTS 13**
 - Identifiers 13
 - Variables 15
 - Assignment Statements 16
 - Tip: Initialize Variables 18
 - More Assignment Statements ★ 19
 - Assignment Compatibility 20
 - Constants 21
 - Arithmetic Operators and Expressions 23
 - Parentheses and Precedence Rules ★ 24
 - Integer and Floating-Point Division 26
 - Pitfall: Round-Off Errors in Floating-Point Numbers 27
 - Pitfall: Division with Whole Numbers 28
 - Type Casting 29
 - Increment and Decrement Operators 30

- 1.3 THE CLASS STRING 33**
 - String Constants and Variables 33
 - Concatenation of Strings 34
 - Classes 35
 - String Methods 37
 - Escape Sequences 42
 - String Processing 43
 - The Unicode Character Set ★ 43

- 1.4 PROGRAM STYLE 46**
 - Naming Constants 46
 - Java Spelling Conventions 48

Comments	49
Indenting	50
Chapter Summary	51
Answers to Self-Test Exercises	52
Programming Projects	54

Chapter 2 Console Input and Output 57

2.1 SCREEN OUTPUT	58
System.out.println	58
Tip: Different Approaches to Formatting Output	61
Formatting Output with printf	61
Tip: Formatting Money Amounts with printf	65
Tip: Legacy Code	66
Money Formats Using NumberFormat ★	67
Importing Packages and Classes	70
The DecimalFormat Class ★	72
2.2 CONSOLE INPUT USING THE SCANNER CLASS	76
The Scanner Class	76
Pitfall: Dealing with the Line Terminator, '\n'	83
The Empty String	84
Tip: Prompt for Input	84
Tip: Echo Input	84
Other Input Delimiters	86
Example: Self-Service Check Out	86
Chapter Summary	89
Answers to Self-Test Exercises	89
Programming Projects	91

Chapter 3 Flow of Control 93

3.1 BRANCHING MECHANISM	94
if-else Statements	94
Omitting the else	95
Compound Statements	96
Tip: Placing of Braces	97
Nested Statements	98
Multiway if-else Statement	98
Example: State Income Tax	99
The switch Statement	101
Pitfall: Forgetting a break in a switch Statement	105
The Conditional Operator ★	106

3.2	BOOLEAN EXPRESSIONS	107
	Simple Boolean Expressions	107
	Pitfall: Using = in Place of ==	108
	Pitfall: Using == with Strings	109
	Lexicographic and Alphabetical Order	110
	Building Boolean Expressions	113
	Evaluating Boolean Expressions	114
	Pitfall: Strings of Inequalities	114
	Tip: Naming Boolean Variables	117
	Short-Circuit and Complete Evaluation	118
	Precedence and Associativity Rules	119
3.3	LOOPS	126
	while Statement and do-while Statement	126
	Algorithms and Pseudocode	128
	Example: Averaging a List of Scores	131
	The for Statement	132
	The Comma in for Statements	135
	Tip: Repeat N Times Loops	137
	Pitfall: Extra Semicolon in a for Statement	137
	Pitfall: Infinite Loops	138
	Nested Loops	139
	The break and continue Statements ★	142
	The exit Statement	143
3.4	DEBUGGING	144
	Loop Bugs	144
	Tracing Variables	144
	General Debugging Techniques	145
	Example: Debugging an Input Validation Loop	146
	Preventive Coding	150
	Assertion Checks ★	151
	Chapter Summary	153
	Answers to Self-Test Exercises	154
	Programming Projects	159

Chapter 4 Defining Classes I 163

4.1	CLASS DEFINITIONS	164
	Instance Variables and Methods	167
	More about Methods	170
	Tip: Any Method Can Be Used as a void Method	174
	Local Variables	176
	Blocks	177

Tip: Declaring Variables in a for Statement 178
Parameters of a Primitive Type 178
Pitfall: Use of the Terms “Parameter” and “Argument” 185
Simple Cases with Class Parameters 187
The this Parameter 187
Methods That Return a Boolean Value 189
The Methods equals and toString 192
Recursive Methods 195
Tip: Testing Methods 195

4.2 INFORMATION HIDING AND ENCAPSULATION 197

public and private Modifiers 198
Example: Yet Another Date Class 199
Accessor and Mutator Methods 200
Tip: A Class Has Access to Private Members of All Objects of the Class 205
Tip: Mutator Methods Can Return a Boolean Value ★ 206
Preconditions and Postconditions 207

4.3 OVERLOADING 208

Rules for Overloading 208
Pitfall: Overloading and Automatic Type Conversion 212
Pitfall: You Cannot Overload Based on the Type Returned 214

4.4 CONSTRUCTORS 216

Constructor Definitions 216
Tip: You Can Invoke Another Method in a Constructor 224
Tip: A Constructor Has a this Parameter 224
Tip: Include a No-Argument Constructor 225
Example: The Final Date Class 226
Default Variable Initializations 227
An Alternative Way to Initialize Instance Variables 227
Example: A Pet Record Class 228
The StringTokenizer Class ★ 232

Chapter Summary 237
Answers to Self-Test Exercises 238
Programming Projects 243

Chapter 5 Defining Classes II 247

5.1 STATIC METHODS AND STATIC VARIABLES 249

Static Methods 249
Pitfall: Invoking a Nonstatic Method Within a Static Method 251
Tip: You Can Put a main in Any Class 252
Static Variables 256

The Math Class 261
Wrapper Classes 265
Automatic Boxing and Unboxing 266
Static Methods in Wrapper Classes 268
Pitfall: A Wrapper Class Does Not Have a No-Argument Constructor 271

5.2 REFERENCES AND CLASS PARAMETERS 272

Variables and Memory 273
References 274
Class Parameters 279
Pitfall: Use of = and == with Variables of a Class Type 283
The Constant null 285
The new Operator and Anonymous Objects 286
Pitfall: Null Pointer Exception 286
Example: Another Approach to Keyboard Input ★ 287
Tip: Use Static Imports ★ 289

5.3 USING AND MISUSING REFERENCES 291

Tip: (continued) 291
Example: A Person Class 292
Pitfall: null Can Be an Argument to a Method 297
Copy Constructors 301
Pitfall: Privacy Leaks 303
Mutable and Immutable Classes 307
Tip: Deep Copy versus Shallow Copy 309

5.4 PACKAGES AND JAVADOC 310

Tip: Assume Your Coworkers Are Malicious 310
Packages and import Statements 311
The Package java.lang 312
Package Names and Directories 312
Pitfall: Subdirectories Are Not Automatically Imported 315
The Default Package 315
Specifying a Class Path When You Compile ★ 316
Pitfall: Not Including the Current Directory in Your Class Path 316
Name Clashes ★ 317
Introduction to javadoc ★ 318
Commenting Classes for javadoc ★ 318
Running javadoc ★ 320

Chapter Summary 322
Answers to Self-Test Exercises 323
Programming Projects 327

Chapter 6 Arrays 333

- 6.1 **INTRODUCTION TO ARRAYS 334**
 - Creating and Accessing Arrays 335
 - The Length Instance Variable 338
 - Tip: Use for Loops with Arrays 340
 - Pitfall: Array Indices Always Start with Zero 340
 - Pitfall: Array Index Out of Bounds 340
 - Initializing Arrays 341
 - Pitfall: An Array of Characters Is Not a String 343

- 6.2 **ARRAYS AND REFERENCES 344**
 - Arrays Are Objects 344
 - Array Parameters 346
 - Pitfall: Arrays with a Class Base Type 346
 - Pitfall: Use of = and == with Arrays 348
 - Arguments for the Method `main` ★ 354
 - Methods that Return an Array 355

- 6.3 **PROGRAMMING WITH ARRAYS 356**
 - Partially Filled Arrays 357
 - Example: A Class for Partially Filled Arrays 360
 - Tip: Accessor Methods Need Not Simply Return Instance Variables 364
 - The “for-each” Loop ★ 364
 - Methods with a Variable Number of Parameters ★ 368
 - Example: A String Processing Example ★ 371
 - Privacy Leaks with Array Instance Variables 372
 - Example: Sorting an Array 376
 - Enumerated Types ★ 380
 - Tip: Enumerated Types in `switch` Statements ★ 385

- 6.4 **MULTIDIMENSIONAL ARRAYS 387**
 - Multidimensional Array Basics 387
 - Using the Length Instance Variable 390
 - Ragged Arrays ★ 391
 - Multidimensional Array Parameters and Returned Values 391
 - Example: A Grade Book Class 392

 - Chapter Summary 398
 - Answers to Self-Test Exercises 399
 - Programming Projects 406

Chapter 7 Inheritance 413

7.1 INHERITANCE BASICS 414

Derived Classes 415

Overriding a Method Definition 425

Changing the Return Type of an Overridden Method 425

Changing the Access Permission of an Overridden Method 426

Pitfall: Overriding versus Overloading 427

The super Constructor 428

The this Constructor 430

Tip: An Object of a Derived Class Has More than One Type 431

Pitfall: The Terms *Subclass* and *Superclass* 434

Example: An Enhanced StringTokenizer Class ★ 435

7.2 ENCAPSULATION AND INHERITANCE 438

Pitfall: Use of Private Instance Variables from the Base Class 438

Protected and Package Access 440

Pitfall: Private Methods Are Effectively Not Inherited 440

Pitfall: Forgetting about the Default Package 442

Pitfall: A Restriction on Protected Access ★ 444

7.3 PROGRAMMING WITH INHERITANCE 446

Tip: Static Variables Are Inherited 446

Access to a Redefined Base Method 447

Tip: “is a” Versus “has a” 447

Pitfall: You Cannot Use Multiple supers 448

The Class Object 449

The Right Way to Define equals 450

Tip: getClass Versus instanceof ★ 453

Chapter Summary 458

Answers to Self-Test Exercises 459

Programming Projects 462

Chapter 8 Polymorphism and Abstract Classes 467

8.1 POLYMORPHISM 468

Late Binding 469

The final Modifier 471

Example: Sales Records 472

Late Binding with toString 479

Pitfall: No Late Binding for Static Methods ★ 480

Downcasting and Upcasting 481

Pitfall: Downcasting 485

Tip: Checking to See Whether Downcasting Is Legitimate ★ 485

A First Look at the clone Method 488

Pitfall: Sometimes the clone Method Return Type Is Object 489
Pitfall: Limitations of Copy Constructors ★ 490

8.2 ABSTRACT CLASSES 494

Abstract Classes 494

Pitfall: You Cannot Create Instances of an Abstract Class 499

Tip: An Abstract Class Is a Type 499

Chapter Summary 501

Answers to Self-Test Exercises 501

Programming Projects 503

Chapter 9 Exception Handling 507

9.1 EXCEPTION HANDLING BASICS 509

Example: A Toy Example of Exception Handling 509

try-throw-catch Mechanism 514

Exception Classes 518

Exception Classes from Standard Packages 519

Defining Exception Classes 521

Tip: Preserve getMessage 525

Tip: An Exception Class Can Carry a Message of Any Type 527

Multiple catch Blocks 532

Pitfall: Catch the More Specific Exception First 534

9.2 THROWING EXCEPTIONS IN METHODS 536

Throwing an Exception in a Method 537

Declaring Exceptions in a throws Clause 539

Exceptions to the Catch or Declare Rule 541

throws Clause in Derived Classes 543

When to Use Exceptions 543

Event-Driven Programming ★ 545

9.3 MORE PROGRAMMING TECHNIQUES FOR EXCEPTION HANDLING 546

The finally Block ★ 547

Pitfall: Nested try-catch Blocks 547

Rethrowing an Exception ★ 549

The AssertionError Class ★ 549

Exception Handling with the Scanner Class 549

Tip: Exception Controlled Loops 550

ArrayIndexOutOfBoundsException 552

Chapter Summary 552

Answers to Self-Test Exercises 553

Programming Projects 557

Chapter 10 File I/O 561

10.1 INTRODUCTION TO FILE I/O 562

Streams 563

Text Files and Binary Files 563

10.2 TEXT FILES 564

Writing to a Text File 564

Pitfall: A try Block Is a Block 570

Appending to a Text File 571

Pitfall: Overwriting an Output File 571

Tip: toString Helps with Text File Output 572

Reading from a Text File 574

Reading a Text File Using Scanner 574

Testing for the End of a Text File with Scanner 577

Reading a Text File Using BufferedReader 584

Testing for the End of a Text File with BufferedReader 588

Tip: Reading Numbers with BufferedReader 588

Path Names 590

Nested Constructor Invocations 591

System.in, System.out, and System.err 592

10.3 THE FILE CLASS 594

Programming with the File Class 594

10.4 BINARY FILES ★ 598

Writing Simple Data to a Binary File 599

UTF and writeUTF 603

Reading Simple Data from a Binary File 604

Checking for the End of a Binary File 609

Binary I/O of Objects 610

Pitfall: Checking for the End of a File in the Wrong Way 610

The Serializable Interface 612

Pitfall: Mixing Class Types in the Same File 615

Array Objects in Binary Files 616

10.5 RANDOM ACCESS TO BINARY FILES ★ 618

Reading and Writing to the Same File 618

Pitfall: A RandomAccessFile Need Not Start Empty 624

Chapter Summary 624

Answers to Self-Test Exercises 625

Programming Projects 629

Chapter 11 Recursion 633

11.1 RECURSIVE VOID METHODS 635

- Example: Vertical Numbers 635
- Tracing a Recursive Call 638
- A Closer Look at Recursion 641
- Pitfall: Infinite Recursion 643
- Stacks for Recursion ★ 644
- Pitfall: Stack Overflow ★ 646
- Recursion versus Iteration 646

11.2 RECURSIVE METHODS THAT RETURN A VALUE 647

- General Form for a Recursive Method That Returns a Value 647
- Example: Another Powers Method 648

11.3 THINKING RECURSIVELY 653

- Recursive Design Techniques 653
- Binary Search ★ 654
- Efficiency of Binary Search ★ 660

- Chapter Summary 662
- Answers to Self-Test Exercises 663
- Programming Projects 666

Chapter 12 UML and Patterns 671

12.1 UML 672

- History of UML 673
- UML Class Diagrams 673
- Class Interactions 674
- Inheritance Diagrams 674
- More UML 676

12.2 PATTERNS 677

- Adaptor Pattern ★ 677
- The Model-View-Controller Pattern ★ 678
- Example: A Sorting Pattern 679
- Restrictions on the Sorting Pattern 685
- Efficiency of the Sorting Pattern ★ 685
- Tip: Pragmatics and Patterns 686
- Pattern Formalism 686

- Chapter Summary 687
- Answers to Self-Test Exercises 687
- Programming Projects 689

Chapter 13 Interfaces and Inner Classes 693

13.1 INTERFACES 695

- Interfaces 695
- Abstract Classes Implementing Interfaces 697
- Derived Interfaces 697
- Pitfall: Interface Semantics Are Not Enforced 699
- The Comparable Interface 701
- Example: Using the Comparable Interface 702
- Defined Constants in Interfaces 707
- Pitfall: Inconsistent Interfaces 708
- The Serializable Interface ★ 711
- The Cloneable Interface 711

13.2 SIMPLE USES OF INNER CLASSES 716

- Helping Classes 716
- Tip: Inner and Outer Classes Have
Access to Each Other's Private Members 717
- Example: A Bank Account Class 717
- The .class File for an Inner Class 721

13.3 MORE ABOUT INNER CLASSES 722

- Static Inner Classes 722
- Pitfall: Other Uses of Inner Classes 722
- Public Inner Classes 723
- Tip: Referring to a Method of the Outer Class 725
- Nesting Inner Classes 727
- Inner Classes and Inheritance 727
- Anonymous Classes 728
- Tip: Why Use Inner Classes? 730
- Chapter Summary 731
- Answers to Self-Test Exercises 732
- Programming Projects 736

Chapter 14 Generics and the ArrayList Class 739

14.1 THE ARRAYLIST CLASS 741

- Using the ArrayList Class 742
- Tip: Summary of Adding to an ArrayList 746
- Methods in the Class ArrayList 747
- The "for-each" Loop 751
- Example: Golf Scores 754
- Tip: Use trimToSize to Save Memory 757
- Pitfall: The clone Method Makes a Shallow Copy ★ 757

The Vector Class 758
Parameterized Classes and Generics 759
Pitfall: Nonparameterized ArrayList and Vector Classes 759

14.2 GENERICS 759

Generic Basics 759
Tip: Compile with the -Xlint Option 761
Example: A Generic Class for Ordered Pairs 762
Pitfall: A Generic Constructor Name Has No Type Parameter 765
Pitfall: You Cannot Plug in a Primitive Type for a Type Parameter 766
Pitfall: A Type Parameter Cannot Be Used
 Everywhere a Type Name Can Be Used 766
Pitfall: An Instantiation of a Generic Class Cannot be an Array Base Type 766
Tip: A Class Definition Can Have More Than One Type Parameter 767
Pitfall: A Generic Class Cannot Be an Exception Class 769
Bounds for Type Parameters 770
Tip: Generic Interfaces 772
Generic Methods ★ 772
Inheritance with Generic Classes ★ 774

Chapter Summary 776
Answers to Self-Test Exercises 777
Programming Projects 779

Chapter 15 Linked Data Structures 783

15.1 JAVA LINKED LISTS 786

Example: A Simple Linked List Class 786
Working with Linked Lists 790
Pitfall: Privacy Leaks 795
Node Inner Classes 796
Example: A Generic Linked List 799
Pitfall: Using Node Instead of Node<T> 804
The equals Method for Linked Lists 805

15.2 COPY CONSTRUCTORS AND THE CLONE METHOD ★ 806

Simple Copy Constructors and clone Methods ★ 807
Exceptions ★ 807
Pitfall: The clone Method Is Protected in Object ★ 810
Tip: Use a Type Parameter Bound for a Better clone ★ 811
Example: A Linked List with a Deep Copy clone Method ★ 814

15.3 ITERATORS 818

Tip: Cloning Is an “All or Nothing” Affair 818
Defining an Iterator Class 819
Adding and Deleting Nodes 824

15.4 VARIATIONS ON A LINKED LIST 827

- Doubly Linked List 829
- The Stack Data Structure 838
- The Queue Data Structure 840
- Running Times and Big-O Notation 843
- Efficiency of Linked Lists 848

15.5 HASH TABLES WITH CHAINING 849

- A Hash Function for Strings 850
- Efficiency of Hash Tables 854

15.6 SETS 855

- Fundamental Set Operations 855
- Efficiency of Sets Using Linked Lists 860

15.7 TREES 861

- Tree Properties 862
- Example: A Binary Search Tree Class ★ 864
- Efficiency of Binary Search Trees 870
- Chapter Summary 871
- Answers to Self-Test Exercises 872
- Programming Projects 877

Chapter 16 Collections, Maps and Iterators 881

- Introduction 882
- Prerequisites 882

16.1 COLLECTIONS 882

- Wildcards 884
- The Collection Framework 884
- Pitfall: Optional Operations 890
- Tip: Dealing with All Those Exceptions 891
- Concrete Collection Classes 892
- Differences Between `ArrayList<T>` and `Vector<T>` 901
- Nonparameterized Version of the Collection Framework ★ 901

16.2 MAPS 902

- Pitfall: Omitting the `<T>` 902
- Concrete Map Classes 905

16.3 ITERATORS 909

- The Iterator Concept 909
- The `Iterator<T>` Interface 909
- Tip: For-Each Loops as Iterators 912

List Iterators 913
Pitfall: next Can Return a Reference 915
Tip: Defining Your Own Iterator Classes 917
Chapter Summary 918
Answers to Self-Test Exercises 918
Programming Projects 919

Chapter 17 Swing I 923

17.1 EVENT-DRIVEN PROGRAMMING 925

Events and Listeners 925

17.2 BUTTONS, EVENTS, AND OTHER SWING BASICS 926

Example: A Simple Window 927

Example: (continued) 928

Pitfall: Forgetting to Program the Close-Window Button 932
Buttons 933

Action Listeners and Action Events 934

Pitfall: Changing the Heading for `actionPerformed` 936

Example: A Better Version of Our First Swing GUI 937

Tip: Ending a Swing Program 937

Labels 941

Color 942

Example: A GUI with a Label and Color 943

17.3 CONTAINERS AND LAYOUT MANAGERS 946

Border Layout Managers 946

Flow Layout Managers 949

Grid Layout Managers 951

Panels 955

Example: A Tricolor Built with Panels 956

The `Container` Class 960

Tip: Code a GUI's Look and Actions Separately 963

The Model-View-Controller Pattern ★ 964

17.4 MENUS AND BUTTONS 965

Example: A GUI with a Menu 965

Menu Bars, Menus, and Menu Items 965

Nested Menus ★ 970

The `AbstractButton` Class 970

The `setActionCommand` Method 973

Listeners as Inner Classes ★ 975

- 17.5 TEXT FIELDS AND TEXT AREAS 978**
 - Text Areas and Text Fields 978
 - Tip: Labeling a Text Field 984
 - A Swing Calculator 985
 - Tip: Inputting and Outputting Numbers 985
 - Chapter Summary 991
 - Answers to Self-Test Exercises 992
 - Programming Projects 999

Chapter 18 Swing II 1001

- 18.1 WINDOW LISTENERS 1002**
 - Example: A Window Listener Inner Class 1004
 - The dispose Method 1008
 - Pitfall: Forgetting to Invoke setDefaultCloseOperation 1008
 - The WindowAdapter Class 1009
- 18.2 ICONS AND SCROLL BARS 1010**
 - Icons 1010
 - Scroll Bars 1017
 - Example: Components with Changing Visibility 1023
- 18.3 THE GRAPHICS CLASS 1026**
 - Coordinate System for Graphics Objects 1027
 - The Method paint and the Class Graphics 1027
 - Drawing Ovals 1033
 - Drawing Arcs 1033
 - Rounded Rectangles ★ 1037
 - paintComponent for Panels 1038
 - Action Drawings and repaint 1038
 - Some More Details on Updating a GUI ★ 1044
- 18.4 COLORS 1044**
 - Specifying a Drawing Color 1045
 - Defining Colors 1046
 - Pitfall: Using doubles to Define a Color 1047
 - The JColorChooser Dialog Window 1048
- 18.5 FONTS AND THE DRAWSTRING METHOD 1051**
 - The drawString Method 1051
 - Fonts 1054
 - Chapter Summary 1057
 - Answers to Self-Test Exercises 1058
 - Programming Projects 1061

Chapter 19 Java Never Ends 1065

Introduction 1066

Prerequisites 1066

19.1 MULTITHREADING 1066

Example: A Nonresponsive GUI 1067

Thread.sleep 1067

The getGraphics Method 1071

Fixing a Nonresponsive Program Using Threads 1072

Example: A Multithreaded Program 1072

The Class Thread 1073

The Runnable Interface ★ 1076

19.2 NETWORKING WITH STREAM SOCKETS 1079

Sockets 1079

Sockets and Threading 1083

19.3 JAVABEANS 1084

The Component Model 1084

The JavaBeans Model 1084

19.4 JAVA AND DATABASE CONNECTIONS 1086

SQL 1086

JDBC 1087

Chapter Summary 1088

Answers to Self-Test Exercises 1089

Programming Projects 1089

Chapter 20 Applets and HTML (on CD)

Appendix 1 Keywords 1091

Appendix 2 Precedence and Associativity Rules 1093

Appendix 3 Unicode Character Set 1095

Appendix 4 Format Specifications for printf 1097

Appendix 5 Summary of Classes and Interfaces 1099

Index 1167