

Contents

Chapter 1 Getting Started 33

1.1 INTRODUCTION TO JAVA 34

- Origins of the Java Language ★ 34
- Objects and Methods 35
- Applets ★ 36
- A Sample Java Application Program 37
- Byte-Code and the Java Virtual Machine 40
- Class Loader ★ 42
- Compiling a Java Program or Class 42
- Running a Java Program 43
- TIP: Error Messages 44

1.2 EXPRESSIONS AND ASSIGNMENT STATEMENTS 45

- Identifiers 45
- Variables 47
- Assignment Statements 48
- TIP: Initialize Variables 50
- More Assignment Statements ★ 51
- Assignment Compatibility 52
- Constants 53
- Arithmetic Operators and Expressions 55
- Parentheses and Precedence Rules ★ 56
- Integer and Floating-Point Division 58
- PITFALL: Round-Off Errors in Floating-Point Numbers 59
- PITFALL: Division with Whole Numbers 60
- Type Casting 61
- Increment and Decrement Operators 62

1.3 THE CLASS `String` 65

- String Constants and Variables 65
- Concatenation of Strings 66
- Classes 67
- String Methods 69
- Escape Sequences 74
- String Processing 75
- The Unicode Character Set ★ 75

- 1.4 PROGRAM STYLE 78**
 - Naming Constants 78
 - Java Spelling Conventions 80
 - Comments 81
 - Indenting 82
 - Chapter Summary 83
 - Answers to Self-Test Exercises 84
 - Programming Projects 86

Chapter 2 Console Input and Output 89

- 2.1 SCREEN OUTPUT 90**
 - `System.out.println` 90
 - TIP: Different Approaches to Formatting Output 93
 - Formatting Output with `printf` 93
 - TIP: Formatting Monetary Amounts with `printf` 97
 - TIP: Legacy Code 98
 - Money Formats Using `NumberFormat` ★ 99
 - Importing Packages and Classes 102
 - The `DecimalFormat` Class ★ 104
- 2.2 CONSOLE INPUT USING THE `Scanner` CLASS 108**
 - The `Scanner` Class 108
 - PITFALL: Dealing with the Line Terminator, `'\n'` 115
 - The Empty String 116
 - TIP: Prompt for Input 116
 - TIP: Echo Input 116
 - EXAMPLE: Self-Service Checkout 118
 - Other Input Delimiters 119
- 2.3 INTRODUCTION TO FILE INPUT 121**
 - The `Scanner` Class for Text File Input 121
 - Chapter Summary 124
 - Answers to Self-Test Exercises 124
 - Programming Projects 127

Chapter 3 Flow of Control 131

- 3.1 BRANCHING MECHANISM 132**
 - `if-else` Statements 132
 - Omitting the `else` 133
 - Compound Statements 134
 - TIP: Placing of Braces 135
 - Nested Statements 136

- Multiway if-else Statement 136
- EXAMPLE: State Income Tax 137
- The switch Statement 139
- PITFALL: Forgetting a break in a switch Statement 143
- The Conditional Operator ★ 144

3.2 BOOLEAN EXPRESSIONS 145

- Simple Boolean Expressions 145
- PITFALL: Using = in Place of == 146
- PITFALL: Using == with Strings 147
- Lexicographic and Alphabetical Order 148
- Building Boolean Expressions 151
- PITFALL: Strings of Inequalities 152
- Evaluating Boolean Expressions 152
- TIP: Naming Boolean Variables 155
- Short-Circuit and Complete Evaluation 156
- Precedence and Associativity Rules 157

3.3 LOOPS 164

- while Statement and do-while Statement 164
- Algorithms and Pseudocode 166
- EXAMPLE: Averaging a List of Scores 169
- The for Statement 170
- The Comma in for Statements 173
- TIP: Repeat N Times Loops 175
- PITFALL: Extra Semicolon in a for Statement 175
- PITFALL: Infinite Loops 176
- Nested Loops 177
- The break and continue Statements ★ 180
- The exit Statement 181

3.4 DEBUGGING 182

- Loop Bugs 182
- Tracing Variables 182
- General Debugging Techniques 183
- EXAMPLE: Debugging an Input Validation Loop 184
- Preventive Coding 188
- Assertion Checks ★ 189

3.5 RANDOM NUMBER GENERATION ★ 191

- The Random Object 192
- The Math.random() Method 193
- Chapter Summary 194
- Answers to Self-Test Exercises 194
- Programming Projects 200

Chapter 4 Defining Classes I 205

4.1 CLASS DEFINITIONS 206

- Instance Variables and Methods 209
- More about Methods 212
- TIP: Any Method Can Be Used as a void Method 216
- Local Variables 218
- Blocks 219
- TIP: Declaring Variables in a for Statement 220
- Parameters of a Primitive Type 220
- PITFALL: Use of the Terms “Parameter” and “Argument” 227
- Simple Cases with Class Parameters 229
- The `this` Parameter 229
- Methods That Return a Boolean Value 231
- The Methods `equals` and `toString` 234
- Recursive Methods 237
- TIP: Testing Methods 237

4.2 INFORMATION HIDING AND ENCAPSULATION 239

- `public` and `private` Modifiers 240
- EXAMPLE: Yet Another Date Class 241
- Accessor and Mutator Methods 242
- TIP: A Class Has Access to Private Members of All Objects of the Class 247
- TIP: Mutator Methods Can Return a Boolean Value ★ 248
- Preconditions and Postconditions 249

4.3 OVERLOADING 250

- Rules for Overloading 250
- PITFALL: Overloading and Automatic Type Conversion 254
- PITFALL: You Cannot Overload Based on the Type Returned 256

4.4 CONSTRUCTORS 258

- Constructor Definitions 258
- TIP: You Can Invoke Another Method in a Constructor 266
- TIP: A Constructor Has a `this` Parameter 266
- TIP: Include a No-Argument Constructor 267
- EXAMPLE: The Final Date Class 268
- Default Variable Initializations 269
- An Alternative Way to Initialize Instance Variables 269
- EXAMPLE: A Pet Record Class 270
- The `StringTokenizer` Class ★ 274
- Chapter Summary 279
- Answers to Self-Test Exercises 280
- Programming Projects 285

Chapter 5 Defining Classes II 291

5.1 STATIC METHODS AND STATIC VARIABLES 293

Static Methods 293

PITFALL: Invoking a Nonstatic Method Within a Static Method 295

TIP: You Can Put a main in Any Class 296

Static Variables 300

The Math Class 305

Wrapper Classes 309

Automatic Boxing and Unboxing 310

Static Methods in Wrapper Classes 312

PITFALL: A Wrapper Class Does Not Have a No-Argument Constructor 315

5.2 REFERENCES AND CLASS PARAMETERS 316

Variables and Memory 317

References 318

Class Parameters 323

PITFALL: Use of = and == with Variables of a Class Type 327

The Constant null 329

PITFALL: Null Pointer Exception 330

The new Operator and Anonymous Objects 330

EXAMPLE: Another Approach to Keyboard Input ★ 331

TIP: Use Static Imports ★ 333

5.3 USING AND MISUSING REFERENCES 335

EXAMPLE: A Person Class 336

PITFALL: null Can Be an Argument to a Method 341

Copy Constructors 345

PITFALL: Privacy Leaks 347

Mutable and Immutable Classes 351

TIP: Deep Copy versus Shallow Copy 353

TIP: Assume Your Coworkers Are Malicious 354

5.4 PACKAGES AND javadoc 354

Packages and import Statements 355

The Package java.lang 356

Package Names and Directories 356

PITFALL: Subdirectories Are Not Automatically Imported 359

The Default Package 359

PITFALL: Not Including the Current Directory in Your Class Path 360

Specifying a Class Path When You Compile ★ 360

Name Clashes ★ 361

Introduction to javadoc ★ 362

Commenting Classes for javadoc ★ 362

Running javadoc ★ 364

Chapter Summary	366
Answers to Self-Test Exercises	367
Programming Projects	371

Chapter 6 Arrays 377

6.1 INTRODUCTION TO ARRAYS 378

Creating and Accessing Arrays	379
The <code>length</code> Instance Variable	382
TIP: Use <code>for</code> Loops with Arrays	384
PITFALL: Array Indices Always Start with Zero	384
PITFALL: Array Index Out of Bounds	384
Initializing Arrays	385
PITFALL: An Array of Characters Is Not a String	387

6.2 ARRAYS AND REFERENCES 388

Arrays Are Objects	388
PITFALL: Arrays with a Class Base Type	390
Array Parameters	390
PITFALL: Use of <code>=</code> and <code>==</code> with Arrays	392
Arguments for the Method <code>main</code> ★	397
Methods that Return an Array	399

6.3 PROGRAMMING WITH ARRAYS 400

Partially Filled Arrays	401
EXAMPLE: A Class for Partially Filled Arrays	404
TIP: Accessor Methods Need Not Simply Return Instance Variables	408
The “for-each” Loop ★	408
Methods with a Variable Number of Parameters ★	412
EXAMPLE: A String Processing Example ★	415
Privacy Leaks with Array Instance Variables	416
EXAMPLE: Sorting an Array	420
Enumerated Types ★	424
TIP: Enumerated Types in <code>switch</code> Statements ★	429

6.4 MULTIDIMENSIONAL ARRAYS 431

Multidimensional Array Basics	431
Using the <code>length</code> Instance Variable	434
Ragged Arrays ★	435
Multidimensional Array Parameters and Returned Values	435
EXAMPLE: A Grade Book Class	436
Chapter Summary	442
Answers to Self-Test Exercises	443
Programming Projects	450

Chapter 7 Inheritance 459**7.1 INHERITANCE BASICS 460**

- Derived Classes 461
- Overriding a Method Definition 471
- Changing the Return Type of an Overridden Method 471
- Changing the Access Permission of an Overridden Method 472
- PITFALL: Overriding versus Overloading 473
- The `super` Constructor 474
- The `this` Constructor 476
- TIP: An Object of a Derived Class Has More than One Type 477
- PITFALL: The Terms *Subclass* and *Superclass* 480
- EXAMPLE: An Enhanced `StringTokenizer` Class ★ 481

7.2 ENCAPSULATION AND INHERITANCE 484

- PITFALL: Use of Private Instance Variables from the Base Class 485
- PITFALL: Private Methods Are Effectively Not Inherited 486
- Protected and Package Access 487
- PITFALL: Forgetting about the Default Package 490
- PITFALL: A Restriction on Protected Access ★ 490

7.3 PROGRAMMING WITH INHERITANCE 493

- TIP: Static Variables Are Inherited 493
- TIP: “is a” versus “has a” 493
- Access to a Redefined Base Method 493
- PITFALL: You Cannot Use Multiple `super`s 495
- The Class Object 496
- The Right Way to Define `equals` 497
- TIP: `getClass` versus `instanceof` ★ 499
- Chapter Summary 504
- Answers to Self-Test Exercises 505
- Programming Projects 508

Chapter 8 Polymorphism and Abstract Classes 515**8.1 POLYMORPHISM 516**

- Late Binding 517
- The `final` Modifier 519
- EXAMPLE: Sales Records 520
- Late Binding with `toString` 527
- PITFALL: No Late Binding for Static Methods 528
- Downcasting and Upcasting 529
- PITFALL: Downcasting 533

- TIP: Checking to See Whether Downcasting Is Legitimate ★ 533
- A First Look at the `clone` Method 536
- PITFALL: Sometimes the `clone` Method Return Type Is `Object` 537
- PITFALL: Limitations of Copy Constructors ★ 538

8.2 ABSTRACT CLASSES 541

- Abstract Classes 542
- PITFALL: You Cannot Create Instances of an Abstract Class 546
- TIP: An Abstract Class Is a Type 547
- Chapter Summary 548
- Answers to Self-Test Exercises 548
- Programming Projects 550

Chapter 9 Exception Handling 555

9.1 EXCEPTION HANDLING BASICS 557

- `try-catch` Mechanism 557
- Exception Handling with the `Scanner` Class 559
- TIP: Exception Controlled Loops 560
- Throwing Exceptions 562
- EXAMPLE: A Toy Example of Exception Handling 564
- Exception Classes 569
- Exception Classes from Standard Packages 570
- Defining Exception Classes 572
- TIP: Preserve `getMessage` 576
- TIP: An Exception Class Can Carry a Message of Any Type 578
- Multiple `catch` Blocks 583
- PITFALL: Catch the More Specific Exception First 585

9.2 THROWING EXCEPTIONS IN METHODS 588

- Throwing an Exception in a Method 588
- Declaring Exceptions in a `throws` Clause 590
- Exceptions to the Catch or Declare Rule 593
- `throws` Clause in Derived Classes 594
- When to Use Exceptions 595
- Event-Driven Programming ★ 596

9.3 MORE PROGRAMMING TECHNIQUES FOR EXCEPTION HANDLING 598

- PITFALL: Nested `try-catch` Blocks 598
- The `finally` Block ★ 598
- Rethrowing an Exception ★ 600
- The `AssertionError` Class ★ 600
- `ArrayIndexOutOfBoundsException` 601

Chapter Summary 601
Answers to Self-Test Exercises 602
Programming Projects 606

Chapter 10 File I/O 609

10.1 INTRODUCTION TO FILE I/O 610

Streams 610
Text Files and Binary Files 611

10.2 TEXT FILES 612

Writing to a Text File 612
PITFALL: A try Block Is a Block 618
PITFALL: Overwriting an Output File 618
Appending to a Text File 619
TIP: toString Helps with Text File Output 620
Reading from a Text File 621
Reading a Text File Using Scanner 621
Testing for the End of a Text File with Scanner 624
Reading a Text File Using BufferedReader 631
TIP: Reading Numbers with BufferedReader 635
Testing for the End of a Text File with BufferedReader 635
Path Names 637
Nested Constructor Invocations 638
System.in, System.out, and System.err 639

10.3 THE File CLASS 641

Programming with the File Class 641

10.4 BINARY FILES ★ 645

Writing Simple Data to a Binary File 646
UTF and writeUTF 650
Reading Simple Data from a Binary File 651
Checking for the End of a Binary File 656
PITFALL: Checking for the End of a File in the Wrong Way 657
Binary I/O of Objects 658
The Serializable Interface 659
PITFALL: Mixing Class Types in the Same File 662
Array Objects in Binary Files 662

10.5 RANDOM ACCESS TO BINARY FILES ★ 664

Reading and Writing to the Same File 664
PITFALL: RandomAccessFile Need Not Start Empty 670

Chapter Summary	670
Answers to Self-Test Exercises	671
Programming Projects	675

Chapter 11 Recursion 679

11.1 RECURSIVE void METHODS 681

EXAMPLE: Vertical Numbers	681
Tracing a Recursive Call	684
A Closer Look at Recursion	687
PITFALL: Infinite Recursion	689
Stacks for Recursion ★	690
PITFALL: Stack Overflow ★	692
Recursion versus Iteration	692

11.2 RECURSIVE METHODS THAT RETURN A VALUE 693

General Form for a Recursive Method That Returns a Value	694
EXAMPLE: Another Powers Method	694

11.3 THINKING RECURSIVELY 699

Recursive Design Techniques	699
Binary Search ★	700
Efficiency of Binary Search ★	706
EXAMPLE: Finding a File	708

Chapter Summary	711
Answers to Self-Test Exercises	711
Programming Projects	716

Chapter 12 UML and Patterns 721

12.1 UML 722

History of UML	723
UML Class Diagrams	723
Class Interactions	724
Inheritance Diagrams	724
More UML	726

12.2 PATTERNS ★ 727

Adaptor Pattern ★	727
The Model-View-Controller Pattern ★	728
EXAMPLE: A Sorting Pattern	729
Restrictions on the Sorting Pattern	735
Efficiency of the Sorting Pattern ★	735

TIP: Pragmatics and Patterns	736
Pattern Formalism	736
Chapter Summary	737
Answers to Self-Test Exercises	737
Programming Projects	739

Chapter 13 Interfaces and Inner Classes 743

13.1 INTERFACES 745

Interfaces	745
Abstract Classes Implementing Interfaces	747
Derived Interfaces	747
PITFALL: Interface Semantics Are Not Enforced	749
The Comparable Interface	751
EXAMPLE: Using the Comparable Interface	752
Defined Constants in Interfaces	757
PITFALL: Inconsistent Interfaces	758
The Serializable Interface ★	761
The Cloneable Interface	761

13.2 SIMPLE USES OF INNER CLASSES 766

Helping Classes	766
TIP: Inner and Outer Classes Have Access to Each Other's Private Members	767
EXAMPLE: A Bank Account Class	767
The .class File for an Inner Class	771
PITFALL: Other Uses of Inner Classes	772

13.3 MORE ABOUT INNER CLASSES 772

Static Inner Classes	772
Public Inner Classes	773
TIP: Referring to a Method of the Outer Class	775
Nesting Inner Classes	777
Inner Classes and Inheritance	777
Anonymous Classes	778
TIP: Why Use Inner Classes?	780
Chapter Summary	781
Answers to Self-Test Exercises	781
Programming Projects	786

Chapter 14 Generics and the ArrayList Class 791

14.1 THE ArrayList CLASS 793

Using the ArrayList Class	794
TIP: Summary of Adding to an ArrayList	798

- Methods in the Class `ArrayList` 799
- The “for-each” Loop 802
- EXAMPLE: Golf Scores 805
- TIP: Use `trimToSize` to Save Memory 808
- PITFALL: The `clone` Method Makes a Shallow Copy ★ 808
- The `Vector` Class 809
- Parameterized Classes and Generics 810
- PITFALL: Nonparameterized `ArrayList` and `Vector` Classes 810

14.2 GENERICS 810

- Generic Basics 811
- TIP: Compile with the `-Xlint` Option 813
- EXAMPLE: A Generic Class for Ordered Pairs 813
- PITFALL: A Generic Constructor Name Has No Type Parameter 816
- PITFALL: You Cannot Plug in a Primitive Type for a Type Parameter 817
- PITFALL: A Type Parameter Cannot Be Used Everywhere a Type Name Can Be Used 817
- PITFALL: An Instantiation of a Generic Class Cannot be an Array Base Type 818
- TIP: A Class Definition Can Have More Than One Type Parameter 819
- PITFALL: A Generic Class Cannot Be an Exception Class 820
- Bounds for Type Parameters 821
- TIP: Generic Interfaces 824
- Generic Methods ★ 824
- Inheritance with Generic Classes ★ 826
- Chapter Summary 828
- Answers to Self-Test Exercises 828
- Programming Projects 831

Chapter 15 Linked Data Structures 833

15.1 JAVA LINKED LISTS 836

- EXAMPLE: A Simple Linked List Class 836
- Working with Linked Lists 840
- PITFALL: Privacy Leaks 845
- Node Inner Classes 846
- EXAMPLE: A Generic Linked List 849
- PITFALL: Using `Node` Instead of `Node<T>` 854
- The `equals` Method for Linked Lists 854

15.2 COPY CONSTRUCTORS AND THE `clone` METHOD ★ 856

- Simple Copy Constructors and `clone` Methods ★ 856
- Exceptions ★ 857

PITFALL: The `clone` Method Is Protected in object ★ 859
TIP: Use a Type Parameter Bound for a Better `clone` ★ 860
EXAMPLE: A Linked List with a Deep Copy `clone` Method ★ 864
TIP: Cloning Is an “All or Nothing” Affair 867

15.3 ITERATORS 867

Defining an Iterator Class 868
Adding and Deleting Nodes 873

15.4 VARIATIONS ON A LINKED LIST 878

Doubly Linked List 878
The Stack Data Structure 887
The Queue Data Structure 889
Running Times and Big-*O* Notation 892
Efficiency of Linked Lists 897

15.5 HASH TABLES WITH CHAINING 898

A Hash Function for Strings 899
Efficiency of Hash Tables 902

15.6 SETS 903

Fundamental Set Operations 904
Efficiency of Sets Using Linked Lists 909

15.7 TREES 910

Tree Properties 910
EXAMPLE: A Binary Search Tree Class ★ 913
Efficiency of Binary Search Trees ★ 918
Chapter Summary 919
Answers to Self-Test Exercises 920
Programming Projects 925

Chapter 16 Collections, Maps and Iterators 929

16.1 COLLECTIONS 930

Wildcards 932
The Collection Framework 932
PITFALL: Optional Operations 938
TIP: Dealing with All Those Exceptions 939
Concrete Collection Classes 940
Differences between `ArrayList<T>` and `Vector<T>` 950
Nonparameterized Version of the Collection Framework ★ 950
PITFALL: Omitting the `<T>` 951

16.2 MAPS 951

Concrete Map Classes 954

16.3 ITERATORS 958

The Iterator Concept 958

The Iterator<T> Interface 958

TIP: For-Each Loops as Iterators 961

List Iterators 962

PITFALL: next Can Return a Reference 964

TIP: Defining Your Own Iterator Classes 966

Chapter Summary 967

Answers to Self-Test Exercises 967

Programming Projects 968

Chapter 17 Swing I 973**17.1 EVENT-DRIVEN PROGRAMMING 975**

Events and Listeners 975

17.2 BUTTONS, EVENTS, AND OTHER SWING BASICS 976

EXAMPLE: A Simple Window 977

PITFALL: Forgetting to Program the Close-Window Button 982

Buttons 983

Action Listeners and Action Events 984

PITFALL: Changing the Heading for actionPerformed 986

TIP: Ending a Swing Program 986

EXAMPLE: A Better Version of Our First Swing GUI 987

Labels 990

Color 991

EXAMPLE: A GUI with a Label and Color 992

17.3 CONTAINERS AND LAYOUT MANAGERS 994

Border Layout Managers 995

Flow Layout Managers 998

Grid Layout Managers 999

Panels 1003

EXAMPLE: A Tricolor Built with Panels 1004

The Container Class 1008

TIP: Code a GUI's Look and Actions Separately 1011

The Model-View-Controller Pattern ★ 1012

17.4 MENUS AND BUTTONS 1013

EXAMPLE: A GUI with a Menu 1013

Menu Bars, Menus, and Menu Items 1013

Nested Menus ★ 1018

The `AbstractButton` Class 1018The `setActionCommand` Method 1021

Listeners as Inner Classes ★ 1022

17.5 TEXT FIELDS AND TEXT AREAS 1025

Text Areas and Text Fields 1026

TIP: Labeling a Text Field 1032

TIP: Inputting and Outputting Numbers 1032

A Swing Calculator 1033

Chapter Summary 1038

Answers to Self-Test Exercises 1039

Programming Projects 1045

Chapter 18 Swing II 1049**18.1 WINDOW LISTENERS 1050**

EXAMPLE: A Window Listener Inner Class 1052

The `dispose` Method 1055PITFALL: Forgetting to Invoke `setDefaultCloseOperation` 1056The `WindowAdapter` Class 1056**18.2 ICONS AND SCROLL BARS 1058**

Icons 1058

Scroll Bars 1064

EXAMPLE: Components with Changing Visibility 1069

18.3 THE `Graphics` CLASS 1073

Coordinate System for Graphics Objects 1073

The Method `paint` and the Class `Graphics` 1074

Drawing Ovals 1079

Drawing Arcs 1079

Rounded Rectangles ★ 1083

`paintComponent` for Panels 1084Action Drawings and `repaint` 1084

Some More Details on Updating a GUI ★ 1090

18.4 COLORS 1090

- Specifying a Drawing Color 1091
- Defining Colors 1092
- PITFALL: Using doubles to Define a Color 1093
- The JColorChooser Dialog Window 1094

18.5 FONTS AND THE drawString METHOD 1097

- The drawString Method 1097
- Fonts 1100
- Chapter Summary 1103
- Answers to Self-Test Exercises 1103
- Programming Projects 1107

Chapter 19 Java Never Ends 1111**19.1 MULTITHREADING 1112**

- EXAMPLE: A Nonresponsive GUI 1113
 - Thread.sleep 1113
 - The getGraphics Method 1117
- Fixing a Nonresponsive Program Using Threads 1118
- EXAMPLE: A Multithreaded Program 1118
- The Class Thread 1119
- The Runnable Interface ★ 1122
- Race Conditions and Thread Synchronization ★ 1125

19.2 NETWORKING WITH STREAM SOCKETS 1130

- Sockets 1130
- Sockets and Threading 1134

19.3 JAVABEANS 1135

- The Component Model 1135
- The JavaBeans Model 1136

19.4 JAVA AND DATABASE CONNECTIONS 1137

- Relational Databases 1137
- Java DB and JDBC 1137
- SQL 1139

19.5 WEB PROGRAMMING WITH JAVA SERVER PAGES 1150

- Applets, Servlets, and Java Server Pages 1150
- Oracle GlassFish Enterprise Server 1152
- HTML Forms—the Common Gateway Interface 1153
- JSP Declarations, Expressions, Scriptlets, and Directives 1155

Chapter Summary	1164
Answers to Self-Test Exercises	1165
Programming Projects	1166

Chapter 20	Applets and HTML (online at www.pearsoninternationaleditions.com/savitch)	1169
-------------------	--	-------------

Appendix 1	Keywords	1171
Appendix 2	Precedence and Associativity Rules	1173
Appendix 3	ASCII Character Set	1175
Appendix 4	Format Specifications for <code>printf</code>	1177
Appendix 5	Summary of Classes and Interfaces	1179
	Index	1247