Chapters 26–34 and Appendices F–N are PDF documents posted online at the book’s Companion Website (located at www.pearsonhighered.com/deitel/). See the inside front cover for information on accessing the Companion Website.

Foreword xxiii

Preface xxv

Before You Begin xxxix

I Introduction to Computers, the Internet and Java 1

1.1 Introduction 2

1.2 Hardware and Software 4

1.2.1 Moore’s Law 4

1.2.2 Computer Organization 5

1.3 Data Hierarchy 6

1.4 Machine Languages, Assembly Languages and High-Level Languages 9

1.5 Introduction to Object Technology 10

1.5.1 The Automobile as an Object 10

1.5.2 Methods and Classes 11

1.5.3 Instantiation 11

1.5.4 Reuse 11

1.5.5 Messages and Method Calls 11

1.5.6 Attributes and Instance Variables 11

1.5.7 Encapsulation and Information Hiding 12

1.5.8 Inheritance 12

1.5.9 Interfaces 12

1.5.10 Object-Oriented Analysis and Design (OOAD) 12

1.5.11 The UML (Unified Modeling Language) 13

1.6 Operating Systems 13

1.6.1 Windows—A Proprietary Operating System 13

1.6.2 Linux—An Open-Source Operating System 14

1.6.3 Android 14

1.7 Programming Languages 15

1.8 Java 17

1.9 A Typical Java Development Environment 17

1.10 Test-Driving a Java Application 21
## Contents

1.11 Internet and World Wide Web  
  1.11.1 The Internet: A Network of Networks  
  1.11.2 The World Wide Web: Making the Internet User-Friendly  
  1.11.3 Web Services and Mashups  
  1.11.4 Ajax  
  1.11.5 The Internet of Things  
1.12 Software Technologies  
1.13 Keeping Up-to-Date with Information Technologies  

2 Introduction to Java Applications; Input/Output and Operators  
  2.1 Introduction  
  2.2 Your First Program in Java: Printing a Line of Text  
  2.3 Modifying Your First Java Program  
  2.4 Displaying Text with `printf`  
  2.5 Another Application: Adding Integers  
    2.5.1 `import` Declarations  
    2.5.2 Declaring Class `Addition`  
    2.5.3 Declaring and Creating a `Scanner` to Obtain User Input from the Keyboard  
    2.5.4 Declaring Variables to Store Integers  
    2.5.5 Prompting the User for Input  
    2.5.6 Obtaining an `int` as Input from the User  
    2.5.7 Prompting for and Inputting a Second `int`  
    2.5.8 Using Variables in a Calculation  
    2.5.9 Displaying the Result of the Calculation  
    2.5.10 Java API Documentation  
  2.6 Memory Concepts  
  2.7 Arithmetic  
  2.8 Decision Making: Equality and Relational Operators  
  2.9 Wrap-Up  

3 Introduction to Classes, Objects, Methods and Strings  
  3.1 Introduction  
  3.2 Instance Variables, `set` Methods and `get` Methods  
    3.2.1 `Account` Class with an Instance Variable, a `set` Method and a `get` Method  
    3.2.2 `AccountTest` Class That Creates and Uses an Object of Class `Account`  
    3.2.3 Compiling and Executing an App with Multiple Classes  
    3.2.4 `Account` UML Class Diagram with an Instance Variable and `set` and `get` Methods  
    3.2.5 Additional Notes on Class `AccountTest`  

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.6</td>
<td>Software Engineering with private Instance Variables and public set and get Methods</td>
<td>79</td>
</tr>
<tr>
<td>3.3</td>
<td>Primitive Types vs. Reference Types</td>
<td>80</td>
</tr>
<tr>
<td>3.4</td>
<td>Account Class: Initializing Objects with Constructors</td>
<td>81</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Declaring an Account Constructor for Custom Object Initialization</td>
<td>81</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Class AccountTest: Initializing Account Objects When They're Created</td>
<td>82</td>
</tr>
<tr>
<td>3.5</td>
<td>Account Class with a Balance; Floating-Point Numbers</td>
<td>84</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Account Class with a balance Instance Variable of Type double</td>
<td>85</td>
</tr>
<tr>
<td>3.5.2</td>
<td>AccountTest Class to Use Class Account</td>
<td>86</td>
</tr>
<tr>
<td>3.6</td>
<td>(Optional) GUI and Graphics Case Study: Using Dialog Boxes</td>
<td>90</td>
</tr>
<tr>
<td>3.7</td>
<td>Wrap-Up</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>Control Statements: Part 1; Assignment, ++ and -- Operators</td>
<td>101</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>102</td>
</tr>
<tr>
<td>4.2</td>
<td>Algorithms</td>
<td>102</td>
</tr>
<tr>
<td>4.3</td>
<td>Pseudocode</td>
<td>103</td>
</tr>
<tr>
<td>4.4</td>
<td>Control Structures</td>
<td>103</td>
</tr>
<tr>
<td>4.5</td>
<td>if Single-Selection Statement</td>
<td>105</td>
</tr>
<tr>
<td>4.6</td>
<td>if...else Double-Selection Statement</td>
<td>106</td>
</tr>
<tr>
<td>4.7</td>
<td>Student Class: Nested if...else Statements</td>
<td>111</td>
</tr>
<tr>
<td>4.8</td>
<td>while Repetition Statement</td>
<td>113</td>
</tr>
<tr>
<td>4.9</td>
<td>Formulating Algorithms: Counter-Controlled Repetition</td>
<td>115</td>
</tr>
<tr>
<td>4.10</td>
<td>Formulating Algorithms: Sentinel-Controlled Repetition</td>
<td>119</td>
</tr>
<tr>
<td>4.11</td>
<td>Formulating Algorithms: Nested Control Statements</td>
<td>126</td>
</tr>
<tr>
<td>4.12</td>
<td>Compound Assignment Operators</td>
<td>131</td>
</tr>
<tr>
<td>4.13</td>
<td>Increment and Decrement Operators</td>
<td>131</td>
</tr>
<tr>
<td>4.14</td>
<td>Primitive Types</td>
<td>134</td>
</tr>
<tr>
<td>4.15</td>
<td>(Optional) GUI and Graphics Case Study: Creating Simple Drawings</td>
<td>135</td>
</tr>
<tr>
<td>4.16</td>
<td>Wrap-Up</td>
<td>139</td>
</tr>
<tr>
<td>5</td>
<td>Control Statements: Part 2; Logical Operators</td>
<td>152</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>153</td>
</tr>
<tr>
<td>5.2</td>
<td>Essentials of Counter-Controlled Repetition</td>
<td>153</td>
</tr>
<tr>
<td>5.3</td>
<td>for Repetition Statement</td>
<td>155</td>
</tr>
<tr>
<td>5.4</td>
<td>Examples Using the for Statement</td>
<td>159</td>
</tr>
<tr>
<td>5.5</td>
<td>do...while Repetition Statement</td>
<td>163</td>
</tr>
<tr>
<td>5.6</td>
<td>switch Multiple-Selection Statement</td>
<td>165</td>
</tr>
<tr>
<td>5.7</td>
<td>Class AutoPolicy Case Study: Strings in switch Statements</td>
<td>171</td>
</tr>
<tr>
<td>5.8</td>
<td>break and continue Statements</td>
<td>174</td>
</tr>
<tr>
<td>5.9</td>
<td>Logical Operators</td>
<td>176</td>
</tr>
<tr>
<td>5.10</td>
<td>Structured Programming Summary</td>
<td>182</td>
</tr>
<tr>
<td>5.11</td>
<td>(Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals</td>
<td>187</td>
</tr>
<tr>
<td>5.12</td>
<td>Wrap-Up</td>
<td>190</td>
</tr>
</tbody>
</table>
### 8 Classes and Objects: A Deeper Look

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Introduction</td>
<td>316</td>
</tr>
<tr>
<td>8.2 Time Class Case Study</td>
<td>316</td>
</tr>
<tr>
<td>8.3 Controlling Access to Members</td>
<td>321</td>
</tr>
<tr>
<td>8.4 Referring to the Current Object’s Members with the this Reference</td>
<td>322</td>
</tr>
<tr>
<td>8.5 Time Class Case Study: Overloaded Constructors</td>
<td>324</td>
</tr>
<tr>
<td>8.6 Default and No-Argument Constructors</td>
<td>330</td>
</tr>
<tr>
<td>8.7 Notes on Set and Get Methods</td>
<td>330</td>
</tr>
<tr>
<td>8.8 Composition</td>
<td>332</td>
</tr>
<tr>
<td>8.9 enum Types</td>
<td>335</td>
</tr>
<tr>
<td>8.10 Garbage Collection</td>
<td>337</td>
</tr>
<tr>
<td>8.11 static Class Members</td>
<td>338</td>
</tr>
<tr>
<td>8.12 static Import</td>
<td>342</td>
</tr>
<tr>
<td>8.13 final Instance Variables</td>
<td>343</td>
</tr>
<tr>
<td>8.14 Package Access</td>
<td>344</td>
</tr>
<tr>
<td>8.15 Using BigDecimal for Precise Monetary Calculations</td>
<td>345</td>
</tr>
<tr>
<td>8.16 (Optional) GUI and Graphics Case Study: Using Objects with Graphics</td>
<td>348</td>
</tr>
<tr>
<td>8.17 Wrap-Up</td>
<td>352</td>
</tr>
</tbody>
</table>

### 9 Object-Oriented Programming: Inheritance

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 Introduction</td>
<td>361</td>
</tr>
<tr>
<td>9.2 Superclasses and Subclasses</td>
<td>362</td>
</tr>
<tr>
<td>9.3 protected Members</td>
<td>364</td>
</tr>
<tr>
<td>9.4 Relationship Between Superclasses and Subclasses</td>
<td>365</td>
</tr>
<tr>
<td>9.4.1 Creating and Using a CommissionEmployee Class</td>
<td>365</td>
</tr>
<tr>
<td>9.4.2 Creating and Using a BasePlusCommissionEmployee Class</td>
<td>371</td>
</tr>
<tr>
<td>9.4.3 Creating a CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy</td>
<td>376</td>
</tr>
<tr>
<td>9.4.4 CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables</td>
<td>379</td>
</tr>
<tr>
<td>9.4.5 CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using private Instance Variables</td>
<td>382</td>
</tr>
<tr>
<td>9.5 Constructors in Subclasses</td>
<td>387</td>
</tr>
<tr>
<td>9.6 Class Object</td>
<td>387</td>
</tr>
<tr>
<td>9.7 (Optional) GUI and Graphics Case Study: Displaying Text and Images Using Labels</td>
<td>388</td>
</tr>
<tr>
<td>9.8 Wrap-Up</td>
<td>391</td>
</tr>
</tbody>
</table>

### 10 Object-Oriented Programming: Polymorphism and Interfaces

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Introduction</td>
<td>396</td>
</tr>
<tr>
<td>10.2 Polymorphism Examples</td>
<td>398</td>
</tr>
<tr>
<td>10.3 Demonstrating Polymorphic Behavior</td>
<td>399</td>
</tr>
<tr>
<td>10.4 Abstract Classes and Methods</td>
<td>401</td>
</tr>
</tbody>
</table>
Contents

10.5 Case Study: Payroll System Using Polymorphism 404
  10.5.1 Abstract Superclass Employee 405
  10.5.2 Concrete Subclass SalariedEmployee 407
  10.5.3 Concrete Subclass HourlyEmployee 409
  10.5.4 Concrete Subclass CommissionEmployee 411
  10.5.5 Indirect Concrete Subclass BasePlusCommissionEmployee 413
  10.5.6 Polymorphic Processing, Operator instanceof and Downcasting 414

10.6 Allowed Assignments Between Superclass and Subclass Variables 419

10.7 final Methods and Classes 419

10.8 A Deeper Explanation of Issues with Calling Methods from Constructors 420

10.9 Creating and Using Interfaces 421
  10.9.1 Developing a Payable Hierarchy 422
  10.9.2 Interface Payable 423
  10.9.3 Class Invoice 424
  10.9.4 Modifying Class Employee to Implement Interface Payable 426
  10.9.5 Modifying Class SalariedEmployee for Use in the Payable Hierarchy 428
  10.9.6 Using Interface Payable to Process Invoices and Employees Polymorphically 430

10.10 Java SE 8 Interface Enhancements 432
  10.10.1 default Interface Methods 432
  10.10.2 static Interface Methods 433
  10.10.3 Functional Interfaces 433

10.11 (Optional) GUI and Graphics Case Study: Drawing with Polymorphism 433

10.12 Wrap-Up 436

11 Exception Handling: A Deeper Look 441

11.1 Introduction 442

11.2 Example: Divide by Zero without Exception Handling 443

11.3 Example: Handling ArithmeticExceptions and InputMismatchExceptions 445

11.4 When to Use Exception Handling 451

11.5 Java Exception Hierarchy 451

11.6 finally Block 454

11.7 Stack Unwinding and Obtaining Information from an Exception Object 459

11.8 Chained Exceptions 461

11.9 Declaring New Exception Types 464

11.10 Preconditions and Postconditions 465

11.11 Assertions 465

11.12 try-with-Resources: Automatic Resource Deallocation 467

11.13 Wrap-Up 467

12 GUI Components: Part I 473

12.1 Introduction 474
## Contents

12.2 Java’s Nimbus Look-and-Feel 475
12.3 Simple GUI-Based Input/Output with JOptionPane 476
12.4 Overview of Swing Components 479
12.5 Displaying Text and Images in a Window 481
12.6 Text Fields and an Introduction to Event Handling with Nested Classes 485
12.7 Common GUI Event Types and Listener Interfaces 491
12.8 How Event Handling Works 493
12.9 JButton 495
12.10 Buttons That Maintain State 498
   12.10.1 JCheckBox 499
   12.10.2 JRadioButton 501
12.11 JComboBox; Using an Anonymous Inner Class for Event Handling 504
12.12 JList 508
12.13 Multiple-Selection Lists 511
12.14 Mouse Event Handling 513
12.15 Adapter Classes 518
12.16 JPanel Subclass for Drawing with the Mouse 522
12.17 Key Event Handling 525
12.18 Introduction to Layout Managers 528
   12.18.1 FlowLayout 530
   12.18.2 BorderLayout 532
   12.18.3 GridLayout 536
12.19 Using Panels to Manage More Complex Layouts 538
12.20 JTextArea 539
12.21 Wrap-Up 542

13 Graphics and Java 2D 555

14 Strings, Characters and Regular Expressions 596
Contents

14.3.4 Locating Characters and Substrings in Strings 605
14.3.5 Extracting Substrings from Strings 607
14.3.6 Concatenating Strings 608
14.3.7 Miscellaneous String Methods 608
14.3.8 String Method valueOf 610
14.4 Class StringBuilder 611
   14.4.1 StringBuilder Constructors 612
   14.4.2 StringBuilder Methods length, capacity, setLength and ensureCapacity 612
   14.4.3 StringBuilder Methods charAt, setCharAt, getChars and reverse 614
   14.4.4 StringBuilder append Methods 615
   14.4.5 StringBuilder Insertion and Deletion Methods 617
14.5 Class Character 618
14.6 Tokenizing Strings 623
14.7 Regular Expressions, Class Pattern and Class Matcher 624
14.8 Wrap-Up 633

15 Files, Streams and Object Serialization 644
15.1 Introduction 645
15.2 Files and Streams 645
15.3 Using NIO Classes and Interfaces to Get File and Directory Information 647
15.4 Sequential-Access Text Files 651
   15.4.1 Creating a Sequential-Access Text File 651
   15.4.2 Reading Data from a Sequential-Access Text File 655
   15.4.3 Case Study: A Credit-Inquiry Program 657
   15.4.4 Updating Sequential-Access Files 661
15.5 Object Serialization 662
   15.5.1 Creating a Sequential-Access File Using Object Serialization 663
   15.5.2 Reading and Deserializing Data from a Sequential-Access File 668
15.6 Opening Files with JFileChooser 670
15.7 (Optional) Additional java.io Classes 673
   15.7.1 Interfaces and Classes for Byte-Based Input and Output 673
   15.7.2 Interfaces and Classes for Character-Based Input and Output 675
15.8 Wrap-Up 676

16 Generic Collections 684
16.1 Introduction 685
16.2 Collections Overview 685
16.3 Type-Wrapper Classes 687
16.4 Autoboxing and Auto-Unboxing 687
16.5 Interface Collection and Class Collections 687
16.6 Lists 688
   16.6.1 ArrayList and Iterator 689
   16.6.2 LinkedList 691
Contents

16.7 Collections Methods  696
  16.7.1 Method sort  697
  16.7.2 Method shuffle  700
  16.7.3 Methods reverse, fill, copy, max and min  702
  16.7.4 Method binarySearch  704
  16.7.5 Methods addAll, frequency and disjoint  706
16.8 Stack Class of Package java.util  708
16.9 Class PriorityQueue and Interface Queue  710
16.10 Sets  711
16.11 Maps  714
16.12 Properties Class  718
16.13 Synchronized Collections  721
16.14 Unmodifiable Collections  721
16.15 Abstract Implementations  722
16.16 Wrap-Up  722

17 Java SE 8 Lambdas and Streams  729

17.1 Introduction  730
17.2 Functional Programming Technologies Overview  731
  17.2.1 Functional Interfaces  732
  17.2.2 Lambda Expressions  733
  17.2.3 Streams  734
17.3 IntStream Operations  736
  17.3.1 Creating an IntStream and Displaying Its Values with the forEach Terminal Operation  738
  17.3.2 Terminal Operations count, min, max, sum and average  739
  17.3.3 Terminal Operation reduce  739
  17.3.4 Intermediate Operations: Filtering and Sorting IntStream Values  741
  17.3.5 Intermediate Operation: Mapping  742
  17.3.6 Creating Streams of ints with IntStream Methods range and rangeClosed  743
17.4 Stream<Integer> Manipulations  743
  17.4.1 Creating a Stream<Integer>  744
  17.4.2 Sorting a Stream and Collecting the Results  745
  17.4.3 Filtering a Stream and Storing the Results for Later Use  745
  17.4.4 Filtering and Sorting a Stream and Collecting the Results  745
  17.4.5 Sorting Previously Collected Results  745
17.5 Stream<String> Manipulations  746
  17.5.1 Mapping Strings to Uppercase Using a Method Reference  747
  17.5.2 Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order  748
  17.5.3 Filtering Strings Then Sorting Them in Case-Insensitive Descending Order  748
17.6 Stream<Employee> Manipulations  748
  17.6.1 Creating and Displaying a List<Employee>  750
### Contents

17.6.2 Filtering Employees with Salaries in a Specified Range 751  
17.6.3 Sorting Employees By Multiple Fields 752  
17.6.4 Mapping Employees to Unique Last Name Strings 754  
17.6.5 Grouping Employees By Department 755  
17.6.6 Counting the Number of Employees in Each Department 756  
17.6.7 Summing and Averaging Employee Salaries 756

17.7 Creating a `Stream<String>` from a File 758  
17.8 Generating Streams of Random Values 761  
17.9 Lambda Event Handlers 763  
17.10 Additional Notes on Java SE 8 Interfaces 763

17.11 Java SE 8 and Functional Programming Resources 764  
17.12 Wrap-Up 764

### 18 Recursion

18.1 Introduction 776  
18.2 Recursion Concepts 777  
18.3 Example Using Recursion: Factorials 778  
18.4 Reimplementing Class `FactorialCalculator` Using Class `BigInteger` 781  
18.5 Example Using Recursion: Fibonacci Series 783  
18.6 Recursion and the Method-Call Stack 786  
18.7 Recursion vs. Iteration 787  
18.8 Towers of Hanoi 789  
18.9 Fractals 791  
18.9.1 Koch Curve Fractal 791  
18.9.2 (Optional) Case Study: Lo Feather Fractal 792

18.10 Recursive Backtracking 801  
18.11 Wrap-Up 802

### 19 Searching, Sorting and Big O

19.1 Introduction 810  
19.2 Linear Search 811  
19.3 Big O Notation 812  
19.3.1 O(1) Algorithms 814  
19.3.2 O(n) Algorithms 815  
19.3.3 O(n^2) Algorithms 815  
19.3.4 Big O of the Linear Search 816  
19.4 Binary Search 816  
19.4.1 Binary Search Implementation 817  
19.4.2 Efficiency of the Binary Search 820  
19.5 Sorting Algorithms 820  
19.6 Selection Sort 821  
19.6.1 Selection Sort Implementation 821  
19.6.2 Efficiency of the Selection Sort 824  
19.7 Insertion Sort 824
## 19.7 Insertion Sort Implementation

19.7.1 Insertion Sort Implementation  825  
19.7.2 Efficiency of the Insertion Sort  827

## 19.8 Merge Sort

19.8.1 Merge Sort Implementation  828  
19.8.2 Efficiency of the Merge Sort  832

## 19.9 Big O Summary for This Chapter’s Searching and Sorting Algorithms  833

## 19.10 Wrap-Up  834

## 20 Generic Classes and Methods  839

20.1 Introduction  840  
20.2 Motivation for Generic Methods  840  
20.3 Generic Methods: Implementation and Compile-Time Translation  842  
20.4 Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type  845  
20.5 Overloading Generic Methods  848  
20.6 Generic Classes  849  
20.7 Raw Types  856  
20.8 Wildcards in Methods That Accept Type Parameters  860  
20.9 Wrap-Up  864

## 21 Custom Generic Data Structures  869

21.1 Introduction  870  
21.2 Self-Referential Classes  871  
21.3 Dynamic Memory Allocation  871  
21.4 Linked Lists  872  
21.4.1 Singly Linked Lists  872  
21.4.2 Implementing a Generic ListNode Class  873  
21.4.3 Generic Classes ListNode and List  878  
21.4.4 Class ListTest  878  
21.4.5 List Method insertAtFront  878  
21.4.6 List Method insertAtBack  879  
21.4.7 List Method removeFromFront  880  
21.4.8 List Method removeFromBack  881  
21.4.9 List Method print  882  
21.4.10 Creating Your Own Packages  882  
21.5 Stacks  886  
21.6 Queues  890  
21.7 Trees  893  
21.8 Wrap-Up  900

## 22 GUI Components: Part 2  911

22.1 Introduction  912  
22.2 JSlider  912
Contents

22.3 Understanding Windows in Java 916
22.4 Using Menus with Frames 917
22.5 JPopupMenu 925
22.6 Pluggable Look-and-Feel 928
22.7 JDesktopPane and JInternalFrame 933
22.8 JTabbedPane 936
22.9 BoxLayout Layout Manager 938
22.10 GridBagLayout Layout Manager 942
22.11 Wrap-Up 952

23 Concurrency 957
23.1 Introduction 958
23.2 Thread States and Life Cycle 960
  23.2.1 New and Runnable States 961
  23.2.2 Waiting State 961
  23.2.3 Timed Waiting State 961
  23.2.4 Blocked State 961
  23.2.5 Terminated State 961
  23.2.6 Operating-System View of the Runnable State 962
  23.2.7 Thread Priorities and Thread Scheduling 962
  23.2.8 Indefinite Postponement and Deadlock 963
23.3 Creating and Executing Threads with the Executor Framework 963
23.4 Thread Synchronization 967
  23.4.1 Immutable Data 968
  23.4.2 Monitors 968
  23.4.3 Unsynchronized Mutable Data Sharing 969
  23.4.4 Synchronized Mutable Data Sharing—Making Operations Atomic 974
23.5 Producer/Consumer Relationship without Synchronization 976
23.6 Producer/Consumer Relationship: ArrayBlockingQueue 984
23.7 (Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll 987
23.8 (Advanced) Producer/Consumer Relationship: Bounded Buffers 994
23.9 (Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces 1002
23.10 Concurrent Collections 1009
23.11 Multithreading with GUI: SwingWorker 1011
  23.11.1 Performing Computations in a Worker Thread: Fibonacci Numbers 1012
  23.11.2 Processing Intermediate Results: Sieve of Eratosthenes 1018
23.12 sort and parallelSort Timings with the Java SE 8 Date/Time API 1025
23.13 Java SE 8: Sequential vs. Parallel Streams 1027
23.14 (Advanced) Interfaces Callable and Future 1030
23.15 (Advanced) Fork/Join Framework 1034
23.16 Wrap-Up 1034

31 JavaServer™ Faces Web Apps: Part 2
32 REST-Based Web Services
33 (Optional) ATM Case Study, Part 1: Object-Oriented Design with the UML
34 (Optional) ATM Case Study, Part 2: Implementing an Object-Oriented Design
F Using the Java API Documentation
G Creating Documentation with javadoc
H Unicode®
I Formatted Output
J Number Systems
K Bit Manipulation
L Labeled break and continue Statements
M UML 2: Additional Diagram Types
N Design Patterns