Contents

Foreword xxiii
Preface xxv
Before You Begin xxxvii

1 Introduction to Java and Test-Driving a Java Application 1
1.1 Introduction 2
1.2 Object Technology Concepts 4
1.2.1 The Automobile as an Object 4
1.2.2 Methods and Classes 4
1.2.3 Instantiation 4
1.2.4 Reuse 5
1.2.5 Messages and Method Calls 5
1.2.6 Attributes and Instance Variables 5
1.2.7 Encapsulation and Information Hiding 5
1.2.8 Inheritance 5
1.2.9 Interfaces 6
1.2.10 Object-Oriented Analysis and Design (OOAD) 6
1.2.11 The UML (Unified Modeling Language) 6
1.3 Open Source Software 7
1.4 Java 8
1.5 A Typical Java Development Environment 9
1.6 Test-Driving a Java Application 12
1.7 Software Technologies 16
1.8 Keeping Up-to-Date with Information Technologies 18

2 Introduction to Java Applications; Input/Output and Operators 20
2.1 Introduction 21
2.2 Your First Program in Java: Printing a Line of Text 21
2.3 Modifying Your First Java Program 26
2.4 Displaying Text with printf 28
## Contents

2.5 Another Application: Adding Integers 29  
2.6 Arithmetic 33  
2.7 Decision Making: Equality and Relational Operators 34  
2.8 Wrap-Up 37  

### 3 Introduction to Classes, Objects, Methods and Strings 38  
3.1 Introduction 39  
3.2 Instance Variables, *set* Methods and *get* Methods 39  
  3.2.1 Account Class with an Instance Variable, a *set* Method and a *get* Method 40  
  3.2.2 AccountTest Class That Creates and Uses an Object of Class Account 43  
  3.2.3 Compiling and Executing an App with Multiple Classes 46  
  3.2.4 Account UML Class Diagram with an Instance Variable and *set* and *get* Methods 46  
  3.2.5 Additional Notes on Class AccountTest 47  
  3.2.6 Software Engineering with private Instance Variables and public *set* and *get* Methods 48  
3.3 Primitive Types vs. Reference Types 49  
3.4 Account Class: Initializing Objects with Constructors 50  
  3.4.1 Declaring an Account Constructor for Custom Object Initialization 50  
  3.4.2 Class AccountTest: Initializing Account Objects When They’re Created 51  
3.5 Account Class with a Balance; Floating-Point Numbers 53  
  3.5.1 Account Class with a balance Instance Variable of Type `double` 54  
  3.5.2 AccountTest Class to Use Class Account 55  
3.6 Wrap-Up 58  

### 4 Control Statements: Part I; Assignment, ++ and -- Operators 59  
4.1 Introduction 60  
4.2 Control Structures 60  
4.3 *if* Single-Selection Statement 62  
4.4 *if...else* Double-Selection Statement 63  
4.5 Student Class: Nested *if...else* Statements 67  
4.6 *while* Repetition Statement 69  
4.7 Counter-Controlled Repetition 71  
4.8 Sentinel-Controlled Repetition 74  
4.9 Nested Control Statements 79  
4.10 Compound Assignment Operators 81  
4.11 Increment and Decrement Operators 81  
4.12 Primitive Types 84  
4.13 Wrap-Up 85
### 5 Control Statements: Part 2; Logical Operators

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>Essentials of Counter-Controlled Repetition</td>
<td>87</td>
</tr>
<tr>
<td>5.3</td>
<td>for Repetition Statement</td>
<td>88</td>
</tr>
<tr>
<td>5.4</td>
<td>Examples Using the for Statement</td>
<td>92</td>
</tr>
<tr>
<td>5.5</td>
<td>do...while Repetition Statement</td>
<td>97</td>
</tr>
<tr>
<td>5.6</td>
<td>switch Multiple-Selection Statement</td>
<td>98</td>
</tr>
<tr>
<td>5.7</td>
<td>Class AutoPolicy Case Study: Strings in switch Statements</td>
<td>104</td>
</tr>
<tr>
<td>5.8</td>
<td>break and continue Statements</td>
<td>108</td>
</tr>
<tr>
<td>5.9</td>
<td>Logical Operators</td>
<td>110</td>
</tr>
<tr>
<td>5.10</td>
<td>Wrap-Up</td>
<td>115</td>
</tr>
</tbody>
</table>

### 6 Methods: A Deeper Look

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>118</td>
</tr>
<tr>
<td>6.2</td>
<td>Program Modules in Java</td>
<td>118</td>
</tr>
<tr>
<td>6.3</td>
<td>static Methods, static Fields and Class Math</td>
<td>119</td>
</tr>
<tr>
<td>6.4</td>
<td>Declaring Methods with Multiple Parameters</td>
<td>121</td>
</tr>
<tr>
<td>6.5</td>
<td>Notes on Declaring and Using Methods</td>
<td>124</td>
</tr>
<tr>
<td>6.6</td>
<td>Argument Promotion and Casting</td>
<td>125</td>
</tr>
<tr>
<td>6.7</td>
<td>Java API Packages</td>
<td>127</td>
</tr>
<tr>
<td>6.8</td>
<td>Case Study: Secure Random-Number Generation</td>
<td>128</td>
</tr>
<tr>
<td>6.9</td>
<td>Case Study: A Game of Chance; Introducing enum Types</td>
<td>133</td>
</tr>
<tr>
<td>6.10</td>
<td>Scope of Declarations</td>
<td>138</td>
</tr>
<tr>
<td>6.11</td>
<td>Method Overloading</td>
<td>140</td>
</tr>
<tr>
<td>6.12</td>
<td>Wrap-Up</td>
<td>142</td>
</tr>
</tbody>
</table>

### 7 Arrays and ArrayLists

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>145</td>
</tr>
<tr>
<td>7.2</td>
<td>Arrays</td>
<td>146</td>
</tr>
<tr>
<td>7.3</td>
<td>Declaring and Creating Arrays</td>
<td>147</td>
</tr>
<tr>
<td>7.4</td>
<td>Examples Using Arrays</td>
<td>148</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Creating and Initializing an Array</td>
<td>148</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Using an Array Initializer</td>
<td>149</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Calculating the Values to Store in an Array</td>
<td>150</td>
</tr>
<tr>
<td>7.4.4</td>
<td>Summing the Elements of an Array</td>
<td>152</td>
</tr>
<tr>
<td>7.4.5</td>
<td>Using Bar Charts to Display Array Data Graphically</td>
<td>152</td>
</tr>
<tr>
<td>7.4.6</td>
<td>Using the Elements of an Array as Counters</td>
<td>154</td>
</tr>
<tr>
<td>7.4.7</td>
<td>Using Arrays to Analyze Survey Results</td>
<td>155</td>
</tr>
<tr>
<td>7.5</td>
<td>Exception Handling: Processing the Incorrect Response</td>
<td>157</td>
</tr>
<tr>
<td>7.5.1</td>
<td>The try Statement</td>
<td>157</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Executing the catch Block</td>
<td>157</td>
</tr>
<tr>
<td>7.5.3</td>
<td>toString Method of the Exception Parameter</td>
<td>158</td>
</tr>
<tr>
<td>7.6</td>
<td>Case Study: Card Shuffling and Dealing Simulation</td>
<td>158</td>
</tr>
<tr>
<td>7.7</td>
<td>Enhanced for Statement</td>
<td>163</td>
</tr>
</tbody>
</table>
10 Object-Oriented Programming: Polymorphism and Interfaces

10.1 Introduction
10.2 Polymorphism Examples
10.3 Demonstrating Polymorphic Behavior
10.4 Abstract Classes and Methods
10.5 Case Study: Payroll System Using Polymorphism
  10.5.1 Abstract Superclass Employee
  10.5.2 Concrete Subclass SalariedEmployee
  10.5.3 Concrete Subclass HourlyEmployee
  10.5.4 Concrete Subclass CommissionEmployee
  10.5.5 Indirect Concrete Subclass BasePlusCommissionEmployee
  10.5.6 Polymorphic Processing, Operator instanceof and Downcasting
10.6 Allowed Assignments Between Superclass and Subclass Variables
10.7 final Methods and Classes
10.8 A Deeper Explanation of Issues with Calling Methods from Constructors
10.9 Creating and Using Interfaces
  10.9.1 Developing a Payable Hierarchy
  10.9.2 Interface Payable
  10.9.3 Class Invoice
  10.9.4 Modifying Class Employee to Implement Interface Payable
  10.9.5 Modifying Class SalariedEmployee for Use in the Payable Hierarchy
  10.9.6 Using Interface Payable to Process Invoices and Employees Polymorphically
  10.9.7 Some Common Interfaces of the Java API
10.10 Java SE 8 Interface Enhancements
  10.10.1 default Interface Methods
  10.10.2 static Interface Methods
  10.10.3 Functional Interfaces
10.11 Wrap-Up

11 Exception Handling: A Deeper Look

11.1 Introduction
11.2 Example: Divide by Zero without Exception Handling
11.3 Exception Handling: ArithmeticExceptions and InputMismatchExceptions
11.4 When to Use Exception Handling
11.5 Java Exception Hierarchy
11.6 finally Block
11.7 Stack Unwinding and Obtaining Information from an Exception Object
12 Swing GUI Components: Part I

12.1 Introduction 333
12.2 Java’s Nimbus Look-and-Feel 334
12.3 Simple GUI-Based Input/Output with JOptionPane 335
12.4 Overview of Swing Components 338
12.5 Displaying Text and Images in a Window 340
12.6 Text Fields and an Introduction to Event Handling with Nested Classes 344
12.7 Common GUI Event Types and Listener Interfaces 350
12.8 How Event Handling Works 352
12.9 JButton 354
12.10 Buttons That Maintain State
   12.10.1 JCheckBox 358
   12.10.2 JRadioButton 360
12.11 JComboBox; Using an Anonymous Inner Class for Event Handling 363
12.12 JList 367
12.13 Multiple-Selection Lists 370
12.14 Mouse Event Handling 372
12.15 Adapter Classes 377
12.16 JPanel Subclass for Drawing with the Mouse 381
12.17 Key Event Handling 384
12.18 Introduction to Layout Managers
   12.18.1 FlowLayout 389
   12.18.2 BorderLayout 391
   12.18.3 GridLayout 395
12.19 Using Panels to Manage More Complex Layouts 397
12.20 JTextArea 398
12.21 Wrap-Up 401

13 Graphics and Java 2D

13.1 Introduction 403
13.2 Graphics Contexts and Graphics Objects 405
13.3 Color Control 406
13.4 Manipulating Fonts 413
13.5 Drawing Lines, Rectangles and Ovals 418
13.6 Drawing Arcs 422
13.7 Drawing Polygons and Polylines 425
13.8 Java 2D API 428
13.9 Wrap-Up 435
## 14 Strings, Characters and Regular Expressions  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.1 Introduction</td>
<td>437</td>
</tr>
<tr>
<td>14.2 Fundamentals of Characters and Strings</td>
<td>437</td>
</tr>
<tr>
<td>14.3 Class String</td>
<td>438</td>
</tr>
<tr>
<td>14.3.1 String Constructors</td>
<td>438</td>
</tr>
<tr>
<td>14.3.2 String Methods length, charAt and getChars</td>
<td>439</td>
</tr>
<tr>
<td>14.3.3 Comparing Strings</td>
<td>440</td>
</tr>
<tr>
<td>14.3.4 Locating Characters and Substrings in Strings</td>
<td>445</td>
</tr>
<tr>
<td>14.3.5 Extracting Substrings from Strings</td>
<td>447</td>
</tr>
<tr>
<td>14.3.6 Concatenating Strings</td>
<td>448</td>
</tr>
<tr>
<td>14.3.7 Miscellaneous String Methods</td>
<td>448</td>
</tr>
<tr>
<td>14.3.8 String Method valueOf</td>
<td>450</td>
</tr>
<tr>
<td>14.4 Class StringBuilder</td>
<td>451</td>
</tr>
<tr>
<td>14.4.1 StringBuilder Constructors</td>
<td>452</td>
</tr>
<tr>
<td>14.4.2 StringBuilder Methods length, capacity, setLength and ensureCapacity</td>
<td>452</td>
</tr>
<tr>
<td>14.4.3 StringBuilder Methods charAt, setCharAt, getChars and reverse</td>
<td>454</td>
</tr>
<tr>
<td>14.4.4 StringBuilder append Methods</td>
<td>455</td>
</tr>
<tr>
<td>14.4.5 StringBuilder Insertion and Deletion Methods</td>
<td>457</td>
</tr>
<tr>
<td>14.5 Class Character</td>
<td>458</td>
</tr>
<tr>
<td>14.6 Tokenizing Strings</td>
<td>463</td>
</tr>
<tr>
<td>14.7 Regular Expressions, Class Pattern and Class Matcher</td>
<td>464</td>
</tr>
<tr>
<td>14.8 Wrap-Up</td>
<td>473</td>
</tr>
</tbody>
</table>

## 15 Files, Streams and Object Serialization  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1 Introduction</td>
<td>475</td>
</tr>
<tr>
<td>15.2 Files and Streams</td>
<td>475</td>
</tr>
<tr>
<td>15.3 Using NIO Classes and Interfaces to Get File and Directory Information</td>
<td>477</td>
</tr>
<tr>
<td>15.4 Sequential-Access Text Files</td>
<td>481</td>
</tr>
<tr>
<td>15.4.1 Creating a Sequential-Access Text File</td>
<td>481</td>
</tr>
<tr>
<td>15.4.2 Reading Data from a Sequential-Access Text File</td>
<td>485</td>
</tr>
<tr>
<td>15.4.3 Case Study: A Credit-Inquiry Program</td>
<td>487</td>
</tr>
<tr>
<td>15.4.4 Updating Sequential-Access Files</td>
<td>491</td>
</tr>
<tr>
<td>15.5 Object Serialization</td>
<td>492</td>
</tr>
<tr>
<td>15.5.1 Creating a Sequential-Access File Using Object Serialization</td>
<td>493</td>
</tr>
<tr>
<td>15.5.2 Reading and Deserializing Data from a Sequential-Access File</td>
<td>498</td>
</tr>
<tr>
<td>15.6 Opening Files with JFileChooser</td>
<td>500</td>
</tr>
<tr>
<td>15.7 (Optional) Additional java.io Classes</td>
<td>503</td>
</tr>
<tr>
<td>15.7.1 Interfaces and Classes for Byte-Based Input and Output</td>
<td>503</td>
</tr>
<tr>
<td>15.7.2 Interfaces and Classes for Character-Based Input and Output</td>
<td>505</td>
</tr>
<tr>
<td>15.8 Wrap-Up</td>
<td>506</td>
</tr>
</tbody>
</table>
### 16 Generic Collections

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1 Introduction</td>
<td>508</td>
</tr>
<tr>
<td>16.2 Collections Overview</td>
<td>508</td>
</tr>
<tr>
<td>16.3 Type-Wrapper Classes</td>
<td>510</td>
</tr>
<tr>
<td>16.4 Autoboxing and Auto-Unboxing</td>
<td>510</td>
</tr>
<tr>
<td>16.5 Interface Collection and Class Collections</td>
<td>510</td>
</tr>
<tr>
<td>16.6 Lists</td>
<td>511</td>
</tr>
<tr>
<td>16.6.1 ArrayList and Iterator</td>
<td>512</td>
</tr>
<tr>
<td>16.6.2 LinkedList</td>
<td>514</td>
</tr>
<tr>
<td>16.7 Collections Methods</td>
<td>519</td>
</tr>
<tr>
<td>16.7.1 Method sort</td>
<td>520</td>
</tr>
<tr>
<td>16.7.2 Method shuffle</td>
<td>523</td>
</tr>
<tr>
<td>16.7.3 Methods reverse, fill, copy, max and min</td>
<td>525</td>
</tr>
<tr>
<td>16.7.4 Method binarySearch</td>
<td>527</td>
</tr>
<tr>
<td>16.7.5 Methods addAll, frequency and disjoint</td>
<td>529</td>
</tr>
<tr>
<td>16.8 Stack Class of Package java.util</td>
<td>531</td>
</tr>
<tr>
<td>16.9 Class PriorityQueue and Interface Queue</td>
<td>533</td>
</tr>
<tr>
<td>16.10 Sets</td>
<td>534</td>
</tr>
<tr>
<td>16.11 Maps</td>
<td>537</td>
</tr>
<tr>
<td>16.12 Properties Class</td>
<td>541</td>
</tr>
<tr>
<td>16.13 Synchronized Collections</td>
<td>544</td>
</tr>
<tr>
<td>16.14 Unmodifiable Collections</td>
<td>544</td>
</tr>
<tr>
<td>16.15 Abstract Implementations</td>
<td>545</td>
</tr>
<tr>
<td>16.16 Wrap-Up</td>
<td>545</td>
</tr>
</tbody>
</table>

### 17 Java SE 8 Lambdas and Streams

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1 Introduction</td>
<td>548</td>
</tr>
<tr>
<td>17.2 Functional Programming Technologies Overview</td>
<td>549</td>
</tr>
<tr>
<td>17.2.1 Functional Interfaces</td>
<td>550</td>
</tr>
<tr>
<td>17.2.2 Lambda Expressions</td>
<td>551</td>
</tr>
<tr>
<td>17.2.3 Streams</td>
<td>552</td>
</tr>
<tr>
<td>17.3 IntStream Operations</td>
<td>554</td>
</tr>
<tr>
<td>17.3.1 Creating an IntStream and Displaying Its Values with the forEach Terminal Operation</td>
<td>556</td>
</tr>
<tr>
<td>17.3.2 Terminal Operations count, min, max, sum and average</td>
<td>557</td>
</tr>
<tr>
<td>17.3.3 Terminal Operation reduce</td>
<td>557</td>
</tr>
<tr>
<td>17.3.4 Intermediate Operations: Filtering and Sorting IntStream Values</td>
<td>559</td>
</tr>
<tr>
<td>17.3.5 Intermediate Operation: Mapping</td>
<td>560</td>
</tr>
<tr>
<td>17.3.6 Creating Streams of ints with IntStream Methods range and rangeClosed</td>
<td>561</td>
</tr>
<tr>
<td>17.4 Stream&lt;Integer&gt; Manipulations</td>
<td>561</td>
</tr>
<tr>
<td>17.4.1 Creating a Stream&lt;Integer&gt;</td>
<td>562</td>
</tr>
<tr>
<td>17.4.2 Sorting a Stream and Collecting the Results</td>
<td>563</td>
</tr>
<tr>
<td>17.4.3 Filtering a Stream and Storing the Results for Later Use</td>
<td>563</td>
</tr>
</tbody>
</table>
## Contents

19.9 BoxLayout Layout Manager 638  
19.10 GridBagLayout Layout Manager 642  
19.11 Wrap-Up 652  

### 20 Concurrency 653

20.1 Introduction 654  
20.2 Thread States and Life Cycle 656  
  20.2.1 New and Runnable States 657  
  20.2.2 Waiting State 657  
  20.2.3 Timed Waiting State 657  
  20.2.4 Blocked State 657  
  20.2.5 Terminated State 657  
  20.2.6 Operating-System View of the Runnable State 658  
  20.2.7 Thread Priorities and Thread Scheduling 658  
  20.2.8 Indefinite Postponement and Deadlock 659  
20.3 Creating and Executing Threads with the Executor Framework 659  
20.4 Thread Synchronization 663  
  20.4.1 Immutable Data 664  
  20.4.2 Monitors 664  
  20.4.3 Unsynchronized Mutable Data Sharing 665  
  20.4.4 Synchronized Mutable Data Sharing—Making Operations Atomic 670  
20.5 Producer/Consumer Relationship without Synchronization 672  
20.6 Producer/Consumer Relationship: ArrayBlockingQueue 680  
20.7 (Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll 683  
20.8 (Advanced) Producer/Consumer Relationship: Bounded Buffers 690  
20.9 (Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces 698  
20.10 Concurrent Collections 705  
20.11 Multithreading with GUI: SwingWorker 707  
  20.11.1 Performing Computations in a Worker Thread: Fibonacci Numbers 708  
  20.11.2 Processing Intermediate Results: Sieve of Eratosthenes 714  
20.12 sort/parallelSort Timings with the Java SE 8 Date/Time API 721  
20.13 Java SE 8: Sequential vs. Parallel Streams 723  
20.14 (Advanced) Interfaces Callable and Future 726  
20.15 (Advanced) Fork/Join Framework 730  
20.16 Wrap-Up 730  

### 21 Accessing Databases with JDBC 732

21.1 Introduction 733  
21.2 Relational Databases 734  
21.3 A books Database 735  
21.4 SQL 739
21.4.1 Basic SELECT Query 739
21.4.2 WHERE Clause 740
21.4.3 ORDER BY Clause 742
21.4.4 Merging Data from Multiple Tables: INNER JOIN 743
21.4.5 INSERT Statement 745
21.4.6 UPDATE Statement 746
21.4.7 DELETE Statement 747
21.5 Setting up a Java DB Database 747
21.5.1 Creating the Chapter's Databases on Windows 748
21.5.2 Creating the Chapter's Databases on Mac OS X 749
21.5.3 Creating the Chapter's Databases on Linux 750
21.6 Manipulating Databases with JDBC 750
21.6.1 Connecting to and Querying a Database 750
21.6.2 Querying the books Database 754
21.7 RowSet Interface 767
21.8 PreparedStatements 769
21.9 Stored Procedures 785
21.10 Transaction Processing 785
21.11 Wrap-Up 786

22 JavaFX GUI 787
22.1 Introduction 788
22.2 JavaFX Scene Builder and the NetBeans IDE 789
22.3 JavaFX App Window Structure 790
22.4 Welcome App—Displaying Text and an Image 791
   22.4.1 Creating the App’s Project 791
   22.4.2 NetBeans Projects Window—Viewing the Project Contents 793
   22.4.3 Adding an Image to the Project 794
   22.4.4 Opening JavaFX Scene Builder from NetBeans 794
   22.4.5 Changing to a VBox Layout Container 795
   22.4.6 Configuring the VBox Layout Container 796
   22.4.7 Adding and Configuring a Label 796
   22.4.8 Adding and Configuring an ImageView 796
   22.4.9 Running the Welcome App 797
22.5 Tip Calculator App—Introduction to Event Handling 798
   22.5.1 Test-Driving the Tip Calculator App 799
   22.5.2 Technologies Overview 799
   22.5.3 Building the App's GUI 802
   22.5.4 TipCalculator Class 806
   22.5.5 TipCalculatorController Class 808
22.6 Wrap-Up 813

23 ATM Case Study, Part 1: Object-Oriented Design with the UML 815
23.1 Case Study Introduction 816
23.2 Examining the Requirements Document 816
23.3 Identifying the Classes in a Requirements Document 824
23.4 Identifying Class Attributes 830
23.5 Identifying Objects' States and Activities 835
23.6 Identifying Class Operations 839
23.7 Indicating Collaboration Among Objects 845
23.8 Wrap-Up 852

24 ATM Case Study Part 2: Implementing an Object-Oriented Design 856
24.1 Introduction 857
24.2 Starting to Program the Classes of the ATM System 857
24.3 Incorporating Inheritance and Polymorphism into the ATM System 862
24.4 ATM Case Study Implementation 868
   24.4.1 Class ATM 869
   24.4.2 Class Screen 874
   24.4.3 Class Keypad 875
   24.4.4 Class CashDispenser 876
   24.4.5 Class DepositSlot 877
   24.4.6 Class Account 878
   24.4.7 Class BankDatabase 880
   24.4.8 Class Transaction 883
   24.4.9 Class BalanceInquiry 884
   24.4.10 Class Withdrawal 885
   24.4.11 Class Deposit 889
   24.4.12 Class ATMCaseStudy 892
24.5 Wrap-Up 893

A Operator Precedence Chart 895
B ASCII Character Set 897
C Keywords and Reserved Words 898
D Primitive Types 899
E Using the Debugger 900
   E.1 Introduction 901
   E.2 Breakpoints and the run, stop, cont and print Commands 901
   E.3 The print and set Commands 905
Contents

J Number Systems
  J.1 Introduction 964
  J.2 Abbreviating Binary Numbers as Octal and Hexadecimal Numbers 967
  J.3 Converting Octal and Hexadecimal Numbers to Binary Numbers 968
  J.4 Converting from Binary, Octal or Hexadecimal to Decimal 968
  J.5 Converting from Decimal to Binary, Octal or Hexadecimal 969
  J.6 Negative Binary Numbers: Two’s Complement Notation 971

K Bit Manipulation
  K.1 Introduction 973
  K.2 Bit Manipulation and the Bitwise Operators 973
  K.3 BitSet Class 983

L Labeled break and continue Statements
  L.1 Introduction 987
  L.2 Labeled break Statement 987
  L.3 Labeled continue Statement 988

M UML 2: Additional Diagram Types
  M.1 Introduction 990
  M.2 Additional Diagram Types 990

N Design Patterns
  N.1 Introduction 992
  N.2 Creational, Structural and Behavioral Design Patterns 993
    N.2.1 Creational Design Patterns 994
    N.2.2 Structural Design Patterns 996
    N.2.3 Behavioral Design Patterns 997
    N.2.4 Conclusion 998
  N.3 Design Patterns in Packages java.awt and javax.swing 998
    N.3.1 Creational Design Patterns 999
    N.3.2 Structural Design Patterns 999
    N.3.3 Behavioral Design Patterns 1001
    N.3.4 Conclusion 1005
  N.4 Concurrency Design Patterns 1005
  N.5 Design Patterns Used in Packages java.io and java.net 1006
    N.5.1 Creational Design Patterns 1006
    N.5.2 Structural Design Patterns 1006
    N.5.3 Architectural Patterns 1008
    N.5.4 Conclusion 1010
  N.6 Design Patterns Used in Package java.util 1010
N.6.1 Creational Design Patterns 1010
N.6.2 Behavioral Design Patterns 1010
N.7 Wrap-Up 1011

Index 1013