



Contents

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 **I**

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

1.11	Internet and World Wide Web	25
1.11.1	The Internet: A Network of Networks	26
1.11.2	The World Wide Web: Making the Internet User-Friendly	26
1.11.3	Web Services and Mashups	26
1.11.4	Ajax	27
1.11.5	The Internet of Things	27
1.12	Software Technologies	28
1.13	Keeping Up-to-Date with Information Technologies	30

2 Introduction to Java Applications; Input/Output and Operators 34

2.1	Introduction	35
2.2	Your First Program in Java: Printing a Line of Text	35
2.3	Modifying Your First Java Program	41
2.4	Displaying Text with <code>printf</code>	43
2.5	Another Application: Adding Integers	44
2.5.1	<code>import</code> Declarations	45
2.5.2	Declaring Class Addition	46
2.5.3	Declaring and Creating a Scanner to Obtain User Input from the Keyboard	46
2.5.4	Declaring Variables to Store Integers	47
2.5.5	Prompting the User for Input	48
2.5.6	Obtaining an <code>int</code> as Input from the User	48
2.5.7	Prompting for and Inputting a Second <code>int</code>	49
2.5.8	Using Variables in a Calculation	49
2.5.9	Displaying the Result of the Calculation	49
2.5.10	Java API Documentation	49
2.6	Memory Concepts	50
2.7	Arithmetic	51
2.8	Decision Making: Equality and Relational Operators	54
2.9	(Optional) GUI and Graphics Case Study: Using Dialog Boxes	58
2.10	Wrap-Up	61

3 Control Statements: Part I; Assignment, ++ and -- Operators 72

3.1	Introduction	73
3.2	Algorithms	73
3.3	Pseudocode	74
3.4	Control Structures	74
3.5	<code>if</code> Single-Selection Statement	76
3.6	<code>if...else</code> Double-Selection Statement	77
3.7	<code>while</code> Repetition Statement	82
3.8	Formulating Algorithms: Counter-Controlled Repetition	83
3.9	Formulating Algorithms: Sentinel-Controlled Repetition	87

3.10	Formulating Algorithms: Nested Control Statements	95
3.11	Compound Assignment Operators	99
3.12	Increment and Decrement Operators	100
3.13	Primitive Types	103
3.14	(Optional) GUI and Graphics Case Study: Creating Simple Drawings	103
3.15	Wrap-Up	107

4 Control Statements: Part 2; Logical Operators 121

4.1	Introduction	122
4.2	Essentials of Counter-Controlled Repetition	122
4.3	for Repetition Statement	124
4.4	Examples Using the for Statement	128
4.5	do...while Repetition Statement	132
4.6	switch Multiple-Selection Statement	134
4.7	break and continue Statements	140
4.8	Logical Operators	142
4.9	Structured Programming Summary	147
4.10	(Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals	152
4.11	Wrap-Up	154

5 Methods 164

5.1	Introduction	165
5.2	Program Modules in Java	165
5.3	static Methods, static Variables and Class Math	167
5.4	Declaring Methods	169
5.5	Notes on Declaring and Using Methods	173
5.6	Method-Call Stack and Stack Frames	174
5.7	Argument Promotion and Casting	175
5.8	Java API Packages	176
5.9	Case Study: Secure Random-Number Generation	178
5.10	Case Study: A Game of Chance; Introducing enum Types	183
5.11	Scope of Declarations	187
5.12	Method Overloading	189
5.13	(Optional) GUI and Graphics Case Study: Colors and Filled Shapes	192
5.14	Wrap-Up	195

6 Arrays and ArrayLists 208

6.1	Introduction	209
6.2	Primitive Types vs. Reference Types	210
6.3	Arrays	210
6.4	Declaring and Creating Arrays	212
6.5	Examples Using Arrays	213
6.5.1	Creating and Initializing an Array	213

x Contents

6.5.2	Using an Array Initializer	214
6.5.3	Calculating the Values to Store in an Array	215
6.5.4	Summing the Elements of an Array	217
6.5.5	Using Bar Charts to Display Array Data Graphically	217
6.5.6	Using the Elements of an Array as Counters	219
6.5.7	Using Arrays to Analyze Survey Results	220
6.6	Exception Handling: Processing the Incorrect Response	222
6.6.1	The <code>try</code> Statement	222
6.6.2	Executing the <code>catch</code> Block	222
6.6.3	<code>toString</code> Method of the Exception Parameter	223
6.7	Enhanced <code>for</code> Statement	223
6.8	Passing Arrays to Methods	224
6.9	Pass-By-Value vs. Pass-By-Reference	227
6.10	Multidimensional Arrays	228
6.11	Variable-Length Argument Lists	231
6.12	Using Command-Line Arguments	233
6.13	Class Arrays	235
6.14	Introduction to Collections and Class <code>ArrayList</code>	238
6.15	(Optional) GUI and Graphics Case Study: Drawing Arcs	241
6.16	Wrap-Up	244

7 Introduction to Classes and Objects **265**

7.1	Introduction	266
7.2	Instance Variables, <i>set</i> Methods and <i>get</i> Methods	267
7.2.1	Account Class with an Instance Variable, a <i>set</i> Method and a <i>get</i> Method	267
7.2.2	AccountTest Class That Creates and Uses an Object of Class Account	269
7.2.3	Compiling and Executing an App with Multiple Classes	272
7.2.4	Account UML Class Diagram with an Instance Variable and <i>set</i> and <i>get</i> Methods	272
7.2.5	Additional Notes on This Example	274
7.2.6	Software Engineering with <code>private</code> Instance Variables and <code>public set</code> and <code>get</code> Methods	274
7.3	Default and Explicit Initialization for Instance Variables	276
7.4	Account Class: Initializing Objects with Constructors	276
7.4.1	Declaring an Account Constructor for Custom Object Initialization	276
7.4.2	Class AccountTest: Initializing Account Objects When They're Created	277
7.5	Account Class with a Balance; Floating-Point Numbers	279
7.5.1	Account Class with a <code>balance</code> Instance Variable of Type <code>double</code>	279
7.5.2	AccountTest Class to Use Class Account	281
7.6	Case Study: Card Shuffling and Dealing Simulation	284
7.7	Case Study: Class GradeBook Using an Array to Store Grades	289

7.8	Case Study: Class <code>GradeBook</code> Using a Two-Dimensional Array	294
7.9	Wrap-Up	300
8	Classes and Objects: A Deeper Look	307
8.1	Introduction	308
8.2	Time Class Case Study	308
8.3	Controlling Access to Members	313
8.4	Referring to the Current Object's Members with the <code>this</code> Reference	314
8.5	Time Class Case Study: Overloaded Constructors	316
8.6	Default and No-Argument Constructors	322
8.7	Notes on <i>Set</i> and <i>Get</i> Methods	322
8.8	Composition	324
8.9	<code>enum</code> Types	327
8.10	Garbage Collection	329
8.11	<code>static</code> Class Members	330
8.12	<code>static</code> Import	334
8.13	<code>final</code> Instance Variables	335
8.14	Package Access	336
8.15	Using <code>BigDecimal</code> for Precise Monetary Calculations	337
8.16	(Optional) GUI and Graphics Case Study: Using Objects with Graphics	340
8.17	Wrap-Up	344
9	Object-Oriented Programming: Inheritance	352
9.1	Introduction	353
9.2	Superclasses and Subclasses	354
9.3	<code>protected</code> Members	356
9.4	Relationship Between Superclasses and Subclasses	357
9.4.1	Creating and Using a <code>CommissionEmployee</code> Class	357
9.4.2	Creating and Using a <code>BasePlusCommissionEmployee</code> Class	363
9.4.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	368
9.4.4	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>protected</code> Instance Variables	371
9.4.5	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>private</code> Instance Variables	374
9.5	Constructors in Subclasses	379
9.6	Class <code>Object</code>	379
9.7	(Optional) GUI and Graphics Case Study: Displaying Text and Images Using Labels	380
9.8	Wrap-Up	383
10	Object-Oriented Programming: Polymorphism and Interfaces	387
10.1	Introduction	388

xii Contents

10.2	Polymorphism Examples	390
10.3	Demonstrating Polymorphic Behavior	391
10.4	Abstract Classes and Methods	393
10.5	Case Study: Payroll System Using Polymorphism	396
10.5.1	Abstract Superclass Employee	397
10.5.2	Concrete Subclass SalariedEmployee	399
10.5.3	Concrete Subclass HourlyEmployee	401
10.5.4	Concrete Subclass CommissionEmployee	403
10.5.5	Indirect Concrete Subclass BasePlusCommissionEmployee	405
10.5.6	Polymorphic Processing, Operator instanceof and Downcasting	406
10.6	Allowed Assignments Between Superclass and Subclass Variables	411
10.7	final Methods and Classes	411
10.8	A Deeper Explanation of Issues with Calling Methods from Constructors	412
10.9	Creating and Using Interfaces	413
10.9.1	Developing a Payable Hierarchy	414
10.9.2	Interface Payable	415
10.9.3	Class Invoice	416
10.9.4	Modifying Class Employee to Implement Interface Payable	418
10.9.5	Modifying Class SalariedEmployee for Use in the Payable Hierarchy	420
10.9.6	Using Interface Payable to Process Invoices and Employees Polymorphically	422
10.9.7	Some Common Interfaces of the Java API	423
10.10	Java SE 8 Interface Enhancements	424
10.10.1	default Interface Methods	424
10.10.2	static Interface Methods	425
10.10.3	Functional Interfaces	425
10.11	(Optional) GUI and Graphics Case Study: Drawing with Polymorphism	425
10.12	Wrap-Up	428

11 Exception Handling: A Deeper Look 433

11.1	Introduction	434
11.2	Example: Divide by Zero without Exception Handling	435
11.3	Example: Handling ArithmeticExceptions and InputMismatchExceptions	437
11.4	When to Use Exception Handling	443
11.5	Java Exception Hierarchy	443
11.6	finally Block	446
11.7	Stack Unwinding and Obtaining Information from an Exception Object	451
11.8	Chained Exceptions	453
11.9	Declaring New Exception Types	456
11.10	Preconditions and Postconditions	457
11.11	Assertions	457
11.12	try-with-Resources: Automatic Resource Deallocation	459
11.13	Wrap-Up	459

12 GUI Components: Part I	465
12.1 Introduction	466
12.2 Java's Nimbus Look-and-Feel	467
12.3 Simple GUI-Based Input/Output with JOptionPane	468
12.4 Overview of Swing Components	471
12.5 Displaying Text and Images in a Window	473
12.6 Text Fields and an Introduction to Event Handling with Nested Classes	477
12.7 Common GUI Event Types and Listener Interfaces	483
12.8 How Event Handling Works	485
12.9 JButton	487
12.10 Buttons That Maintain State	490
12.10.1 JCheckBox	491
12.10.2 JRadioButton	493
12.11 JComboBox; Using an Anonymous Inner Class for Event Handling	496
12.12 JList	500
12.13 Multiple-Selection Lists	503
12.14 Mouse Event Handling	505
12.15 Adapter Classes	510
12.16 JPanel Subclass for Drawing with the Mouse	514
12.17 Key Event Handling	517
12.18 Introduction to Layout Managers	520
12.18.1 FlowLayout	522
12.18.2 BorderLayout	524
12.18.3 GridLayout	528
12.19 Using Panels to Manage More Complex Layouts	530
12.20 JTextArea	531
12.21 Wrap-Up	534
13 Graphics and Java 2D	547
13.1 Introduction	548
13.2 Graphics Contexts and Graphics Objects	550
13.3 Color Control	551
13.4 Manipulating Fonts	558
13.5 Drawing Lines, Rectangles and Ovals	563
13.6 Drawing Arcs	567
13.7 Drawing Polygons and Polylines	570
13.8 Java 2D API	573
13.9 Wrap-Up	580
14 Strings, Characters and Regular Expressions	588
14.1 Introduction	589
14.2 Fundamentals of Characters and Strings	589
14.3 Class String	590
14.3.1 String Constructors	590

14.3.2	String Methods <code>length</code> , <code>charAt</code> and <code>getChars</code>	591
14.3.3	Comparing Strings	592
14.3.4	Locating Characters and Substrings in Strings	597
14.3.5	Extracting Substrings from Strings	599
14.3.6	Concatenating Strings	600
14.3.7	Miscellaneous String Methods	600
14.3.8	String Method <code>valueOf</code>	602
14.4	Class <code>StringBuilder</code>	603
14.4.1	<code>StringBuilder</code> Constructors	604
14.4.2	<code>StringBuilder</code> Methods <code>length</code> , <code>capacity</code> , <code>setLength</code> and <code>ensureCapacity</code>	604
14.4.3	<code>StringBuilder</code> Methods <code>charAt</code> , <code>setCharAt</code> , <code>getChars</code> and <code>reverse</code>	606
14.4.4	<code>StringBuilder</code> <code>append</code> Methods	607
14.4.5	<code>StringBuilder</code> Insertion and Deletion Methods	609
14.5	Class <code>Character</code>	610
14.6	Tokenizing Strings	615
14.7	Regular Expressions, Class <code>Pattern</code> and Class <code>Matcher</code>	616
14.8	Wrap-Up	625

15 Files, Streams and Object Serialization 636

15.1	Introduction	637
15.2	Files and Streams	637
15.3	Using NIO Classes and Interfaces to Get File and Directory Information	639
15.4	Sequential-Access Text Files	643
15.4.1	Creating a Sequential-Access Text File	643
15.4.2	Reading Data from a Sequential-Access Text File	647
15.4.3	Case Study: A Credit-Inquiry Program	649
15.4.4	Updating Sequential-Access Files	653
15.5	Object Serialization	654
15.5.1	Creating a Sequential-Access File Using Object Serialization	655
15.5.2	Reading and Deserializing Data from a Sequential-Access File	660
15.6	Opening Files with <code>JFileChooser</code>	662
15.7	(Optional) Additional <code>java.io</code> Classes	665
15.7.1	Interfaces and Classes for Byte-Based Input and Output	665
15.7.2	Interfaces and Classes for Character-Based Input and Output	667
15.8	Wrap-Up	668

16 Generic Collections 676

16.1	Introduction	677
16.2	Collections Overview	677
16.3	Type-Wrapper Classes	679
16.4	Autoboxing and Auto-Unboxing	679
16.5	Interface <code>Collection</code> and Class <code>Collections</code>	679

16.6	Lists	680
16.6.1	ArrayList and Iterator	681
16.6.2	LinkedList	683
16.7	Collections Methods	688
16.7.1	Method sort	689
16.7.2	Method shuffle	692
16.7.3	Methods reverse, fill, copy, max and min	694
16.7.4	Method binarySearch	696
16.7.5	Methods addAll, frequency and disjoint	698
16.8	Stack Class of Package java.util	700
16.9	Class PriorityQueue and Interface Queue	702
16.10	Sets	703
16.11	Maps	706
16.12	Properties Class	710
16.13	Synchronized Collections	713
16.14	Unmodifiable Collections	713
16.15	Abstract Implementations	714
16.16	Wrap-Up	714

17 Java SE 8 Lambdas and Streams 721

17.1	Introduction	722
17.2	Functional Programming Technologies Overview	723
17.2.1	Functional Interfaces	724
17.2.2	Lambda Expressions	725
17.2.3	Streams	726
17.3	IntStream Operations	728
17.3.1	Creating an IntStream and Displaying Its Values with the forEach Terminal Operation	730
17.3.2	Terminal Operations count, min, max, sum and average	731
17.3.3	Terminal Operation reduce	731
17.3.4	Intermediate Operations: Filtering and Sorting IntStream Values	733
17.3.5	Intermediate Operation: Mapping	734
17.3.6	Creating Streams of ints with IntStream Methods range and rangeClosed	735
17.4	Stream<Integer> Manipulations	735
17.4.1	Creating a Stream<Integer>	736
17.4.2	Sorting a Stream and Collecting the Results	737
17.4.3	Filtering a Stream and Storing the Results for Later Use	737
17.4.4	Filtering and Sorting a Stream and Collecting the Results	737
17.4.5	Sorting Previously Collected Results	737
17.5	Stream<String> Manipulations	738
17.5.1	Mapping Strings to Uppercase Using a Method Reference	739
17.5.2	Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order	740

17.5.3	Filtering Strings Then Sorting Them in Case-Insensitive Descending Order	740
17.6	Stream<Employee> Manipulations	740
17.6.1	Creating and Displaying a List<Employee>	742
17.6.2	Filtering Employees with Salaries in a Specified Range	743
17.6.3	Sorting Employees By Multiple Fields	744
17.6.4	Mapping Employees to Unique Last Name Strings	746
17.6.5	Grouping Employees By Department	747
17.6.6	Counting the Number of Employees in Each Department	748
17.6.7	Summing and Averaging Employee Salaries	748
17.7	Creating a Stream<String> from a File	750
17.8	Generating Streams of Random Values	753
17.9	Lambda Event Handlers	755
17.10	Additional Notes on Java SE 8 Interfaces	755
17.11	Java SE 8 and Functional Programming Resources	756
17.12	Wrap-Up	756

18 Recursion 768

18.1	Introduction	769
18.2	Recursion Concepts	770
18.3	Example Using Recursion: Factorials	771
18.4	Reimplementing Class FactorialCalculator Using Class BigInteger	773
18.5	Example Using Recursion: Fibonacci Series	775
18.6	Recursion and the Method-Call Stack	778
18.7	Recursion vs. Iteration	779
18.8	Towers of Hanoi	781
18.9	Fractals	783
18.9.1	Koch Curve Fractal	783
18.9.2	(Optional) Case Study: Lo Feather Fractal	784
18.10	Recursive Backtracking	793
18.11	Wrap-Up	794

19 Searching, Sorting and Big O 802

19.1	Introduction	803
19.2	Linear Search	804
19.3	Big O Notation	806
19.3.1	O(1) Algorithms	806
19.3.2	O(<i>n</i>) Algorithms	807
19.3.3	O(<i>n</i> ²) Algorithms	807
19.3.4	Big O of the Linear Search	808
19.4	Binary Search	808
19.4.1	Binary Search Implementation	809
19.4.2	Efficiency of the Binary Search	812
19.5	Sorting Algorithms	812

19.6	Selection Sort	813
19.6.1	Selection Sort Implementation	813
19.6.2	Efficiency of the Selection Sort	816
19.7	Insertion Sort	816
19.7.1	Insertion Sort Implementation	817
19.7.2	Efficiency of the Insertion Sort	819
19.8	Merge Sort	819
19.8.1	Merge Sort Implementation	820
19.8.2	Efficiency of the Merge Sort	824
19.9	Big O Summary for This Chapter's Searching and Sorting Algorithms	825
19.10	Wrap-Up	826

20 Generic Classes and Methods 831

20.1	Introduction	832
20.2	Motivation for Generic Methods	832
20.3	Generic Methods: Implementation and Compile-Time Translation	834
20.4	Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type	837
20.5	Overloading Generic Methods	840
20.6	Generic Classes	841
20.7	Raw Types	848
20.8	Wildcards in Methods That Accept Type Parameters	852
20.9	Wrap-Up	856

21 Custom Generic Data Structures 861

21.1	Introduction	862
21.2	Self-Referential Classes	863
21.3	Dynamic Memory Allocation	863
21.4	Linked Lists	864
21.4.1	Singly Linked Lists	864
21.4.2	Implementing a Generic List Class	865
21.4.3	Generic Classes ListNode and List	870
21.4.4	Class ListTest	870
21.4.5	List Method insertAtFront	870
21.4.6	List Method insertAtBack	871
21.4.7	List Method removeFromFront	872
21.4.8	List Method removeFromBack	873
21.4.9	List Method print	874
21.4.10	Creating Your Own Packages	874
21.5	Stacks	878
21.6	Queues	882
21.7	Trees	885
21.8	Wrap-Up	892

22	GUI Components: Part 2	903
22.1	Introduction	904
22.2	JSlider	904
22.3	Understanding Windows in Java	908
22.4	Using Menus with Frames	909
22.5	JPopupMenu	917
22.6	Pluggable Look-and-Feel	920
22.7	JDesktopPane and JInternalFrame	925
22.8	JTabbedPane	928
22.9	BoxLayout Layout Manager	930
22.10	GridBagLayout Layout Manager	934
22.11	Wrap-Up	944
23	Concurrency	949
23.1	Introduction	950
23.2	Thread States and Life Cycle	952
23.2.1	<i>New</i> and <i>Runnable</i> States	953
23.2.2	<i>Waiting</i> State	953
23.2.3	<i>Timed Waiting</i> State	953
23.2.4	<i>Blocked</i> State	953
23.2.5	<i>Terminated</i> State	953
23.2.6	Operating-System View of the <i>Runnable</i> State	954
23.2.7	Thread Priorities and Thread Scheduling	954
23.2.8	Indefinite Postponement and Deadlock	955
23.3	Creating and Executing Threads with the Executor Framework	955
23.4	Thread Synchronization	959
23.4.1	Immutable Data	960
23.4.2	Monitors	960
23.4.3	Unsynchronized Mutable Data Sharing	961
23.4.4	Synchronized Mutable Data Sharing—Making Operations Atomic	966
23.5	Producer/Consumer Relationship without Synchronization	968
23.6	Producer/Consumer Relationship: ArrayBlockingQueue	976
23.7	(Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll	979
23.8	(Advanced) Producer/Consumer Relationship: Bounded Buffers	986
23.9	(Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces	994
23.10	Concurrent Collections	1001
23.11	Multithreading with GUI: SwingWorker	1003
23.11.1	Performing Computations in a Worker Thread: Fibonacci Numbers	1004
23.11.2	Processing Intermediate Results: Sieve of Eratosthenes	1010
23.12	sort/parallelSort Timings with the Java SE 8 Date/Time API	1017
23.13	Java SE 8: Sequential vs. Parallel Streams	1019
23.14	(Advanced) Interfaces Callable and Future	1022

23.15 (Advanced) Fork/Join Framework	1026
23.16 Wrap-Up	1026

24 Accessing Databases with JDBC **1037**

24.1 Introduction	1038
24.2 Relational Databases	1039
24.3 A books Database	1040
24.4 SQL	1044
24.4.1 Basic SELECT Query	1044
24.4.2 WHERE Clause	1045
24.4.3 ORDER BY Clause	1047
24.4.4 Merging Data from Multiple Tables: INNER JOIN	1048
24.4.5 INSERT Statement	1050
24.4.6 UPDATE Statement	1051
24.4.7 DELETE Statement	1052
24.5 Setting up a Java DB Database	1052
24.5.1 Creating the Chapter's Databases on Windows	1053
24.5.2 Creating the Chapter's Databases on Mac OS X	1054
24.5.3 Creating the Chapter's Databases on Linux	1055
24.6 Manipulating Databases with JDBC	1055
24.6.1 Connecting to and Querying a Database	1055
24.6.2 Querying the books Database	1059
24.7 RowSet Interface	1072
24.8 PreparedStatement	1074
24.9 Stored Procedures	1090
24.10 Transaction Processing	1090
24.11 Wrap-Up	1091

25 JavaFX GUI: Part I **1099**

25.1 Introduction	1100
25.2 JavaFX Scene Builder and the NetBeans IDE	1101
25.3 JavaFX App Window Structure	1102
25.4 Welcome App—Displaying Text and an Image	1103
25.4.1 Creating the App's Project	1103
25.4.2 NetBeans Projects Window—Viewing the Project Contents	1105
25.4.3 Adding an Image to the Project	1106
25.4.4 Opening JavaFX Scene Builder from NetBeans	1106
25.4.5 Changing to a VBox Layout Container	1107
25.4.6 Configuring the VBox Layout Container	1108
25.4.7 Adding and Configuring a Label	1108
25.4.8 Adding and Configuring an ImageView	1108
25.4.9 Running the Welcome App	1109
25.5 Tip Calculator App—Introduction to Event Handling	1110
25.5.1 Test-Driving the Tip Calculator App	1111

xx Contents

25.5.2	Technologies Overview	1111
25.5.3	Building the App's GUI	1114
25.5.4	TipCalculator Class	1118
25.5.5	TipCalculatorController Class	1120
25.6	Features Covered in the Online JavaFX Chapters	1125
25.7	Wrap-Up	1126

Chapters on the Web **I 133**

A Operator Precedence Chart **I 135**

B ASCII Character Set **I 137**

C Keywords and Reserved Words **I 138**

D Primitive Types **I 139**

E Using the Debugger **I 140**

E.1	Introduction	1141
E.2	Breakpoints and the run, stop, cont and print Commands	1141
E.3	The print and set Commands	1145
E.4	Controlling Execution Using the step, step up and next Commands	1147
E.5	The watch Command	1150
E.6	The clear Command	1152
E.7	Wrap-Up	1154

Appendices on the Web **I 157**

Index **I 159**

Online Chapters and Appendices

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.

26 JavaFX GUI: Part 2

27 JavaFX Graphics and Multimedia

28 Networking

- 29** Java Persistence API (JPA)
- 30** JavaServer™ Faces Web Apps: Part 1
- 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

