

INTRODUCTION TO  
**JAVA**<sup>™</sup>  
PROGRAMMING AND  
DATA STRUCTURES  
COMPREHENSIVE VERSION

Twelfth Edition

**Y. Daniel Liang**

*Georgia Southern University*



# BRIEF CONTENTS

---

1	Introduction to Computers, Programs, and Java™	1	30	Aggregate Operations for Collection Streams	1129	
2	Elementary Programming	33	CHAPTER 31–44 are available from the Companion Website at <a href="http://www.pearsonhighered.com/liang">www.pearsonhighered.com/liang</a>			
3	Selections	77	31	Advanced JavaFX and FXML		
4	Mathematical Functions, Characters, and Strings	121	32	Multithreading and Parallel Programming		
5	Loops	159	33	Networking		
6	Methods	205	34	Java Database Programming		
7	Single-Dimensional Arrays	249	35	Advanced Java Database Programming		
8	Multidimensional Arrays	289	36	Internationalization		
9	Objects and Classes	323	37	Servlets		
10	Object-Oriented Thinking	367	38	JavaServer Pages		
11	Inheritance and Polymorphism	411	39	JavaServer Faces		
12	Exception Handling and Text I/O	453	40	RMI		
13	Abstract Classes and Interfaces	499	41	Web Services		
14	JavaFX Basics	541	42	2-4 Trees and B-Trees		
15	Event-Driven Programming and Animations	593	43	Red-Black Trees		
16	JavaFX UI Controls and Multimedia	643	44	Testing Using JUnit		
17	Binary I/O	691	APPENDIXES			1161
18	Recursion	719	A	Java Keywords and Reserved Words	1163	
19	Generics	751	B	The ASCII Character Set	1164	
20	Lists, Stacks, Queues, and Priority Queues	775	C	Operator Precedence Chart	1166	
21	Sets and Maps	815	D	Java Modifiers	1168	
22	Developing Efficient Algorithms	839	E	Special Floating-Point Values	1170	
23	Sorting	887	F	Number Systems	1171	
24	Implementing Lists, Stacks, Queues, and Priority Queues	923	G	Bitwise Operations	1175	
25	Binary Search Trees	959	H	Regular Expressions	1176	
26	AVL Trees	995	I	Enumerated Types	1182	
27	Hashing	1015	J	The Big-O, Big-Omega, and Big-Theta Notations	1187	
28	Graphs and Applications	1045	QUICK REFERENCE			1189
29	Weighted Graphs and Applications	1091	INDEX			1191

# CONTENTS

---

<b>Chapter 1</b>	<b>Introduction to Computers, Programs, and Java™</b>	<b>1</b>
1.1	Introduction	2
1.2	What Is a Computer?	2
1.3	Programming Languages	7
1.4	Operating Systems	9
1.5	Java, the World Wide Web, and Beyond	10
1.6	The Java Language Specification, API, JDK, JRE, and IDE	11
1.7	A Simple Java Program	12
1.8	Creating, Compiling, and Executing a Java Program	15
1.9	Programming Style and Documentation	18
1.10	Programming Errors	19
1.11	Developing Java Programs Using NetBeans	23
1.12	Developing Java Programs Using Eclipse	26
<b>Chapter 2</b>	<b>Elementary Programming</b>	<b>33</b>
2.1	Introduction	34
2.2	Writing a Simple Program	34
2.3	Reading Input from the Console	37
2.4	Identifiers	40
2.5	Variables	40
2.6	Assignment Statements and Assignment Expressions	42
2.7	Named Constants	43
2.8	Naming Conventions	44
2.9	Numeric Data Types and Operations	45
2.10	Numeric Literals	48
2.11	JShell	50
2.12	Evaluating Expressions and Operator Precedence	52
2.13	Case Study: Displaying the Current Time	54
2.14	Augmented Assignment Operators	56
2.15	Increment and Decrement Operators	57
2.16	Numeric Type Conversions	58
2.17	Software Development Process	61
2.18	Case Study: Counting Monetary Units	64
2.19	Common Errors and Pitfalls	67
<b>Chapter 3</b>	<b>Selections</b>	<b>77</b>
3.1	Introduction	78
3.2	boolean Data Type, Values, and Expressions	78
3.3	if Statements	80
3.4	Two-Way if-else Statements	82
3.5	Nested if and Multi-Way if-else Statements	83
3.6	Common Errors and Pitfalls	85
3.7	Generating Random Numbers	89
3.8	Case Study: Computing Body Mass Index	91
3.9	Case Study: Computing Taxes	92
3.10	Logical Operators	95
3.11	Case Study: Determining Leap Year	99
3.12	Case Study: Lottery	100

3.13	switch Statements	102
3.14	Conditional Operators	105
3.15	Operator Precedence and Associativity	106
3.16	Debugging	108
<b>Chapter 4</b>	<b>Mathematical Functions, Characters, and Strings</b>	<b>121</b>
4.1	Introduction	122
4.2	Common Mathematical Functions	122
4.3	Character Data Type and Operations	126
4.4	The String Type	131
4.5	Case Studies	140
4.6	Formatting Console Output	146
<b>Chapter 5</b>	<b>Loops</b>	<b>159</b>
5.1	Introduction	160
5.2	The while Loop	160
5.3	Case Study: Guessing Numbers	163
5.4	Loop Design Strategies	166
5.5	Controlling a Loop with User Confirmation or a Sentinel Value	168
5.6	The do-while Loop	171
5.7	The for Loop	173
5.8	Which Loop to Use?	176
5.9	Nested Loops	178
5.10	Minimizing Numeric Errors	180
5.11	Case Studies	182
5.12	Keywords <i>break</i> and <i>continue</i>	186
5.13	Case Study: Checking Palindromes	189
5.14	Case Study: Displaying Prime Numbers	191
<b>Chapter 6</b>	<b>Methods</b>	<b>205</b>
6.1	Introduction	206
6.2	Defining a Method	206
6.3	Calling a Method	208
6.4	void vs. Value-Returning Methods	211
6.5	Passing Arguments by Values	213
6.6	Modularizing Code	217
6.7	Case Study: Converting Hexadecimals to Decimals	219
6.8	Overloading Methods	221
6.9	The Scope of Variables	224
6.10	Case Study: Generating Random Characters	225
6.11	Method Abstraction and Stepwise Refinement	227
<b>Chapter 7</b>	<b>Single-Dimensional Arrays</b>	<b>249</b>
7.1	Introduction	250
7.2	Array Basics	250
7.3	Case Study: Analyzing Numbers	257
7.4	Case Study: Deck of Cards	258
7.5	Copying Arrays	260
7.6	Passing Arrays to Methods	261
7.7	Returning an Array from a Method	264
7.8	Case Study: Counting the Occurrences of Each Letter	265
7.9	Variable-Length Argument Lists	268
7.10	Searching Arrays	269

7.11	Sorting Arrays	273
7.12	The Arrays Class	274
7.13	Command-Line Arguments	276
<b>Chapter 8</b>	<b>Multidimensional Arrays</b>	<b>289</b>
8.1	Introduction	290
8.2	Two-Dimensional Array Basics	290
8.3	Processing Two-Dimensional Arrays	293
8.4	Passing Two-Dimensional Arrays to Methods	295
8.5	Case Study: Grading a Multiple-Choice Test	296
8.6	Case Study: Finding the Closest Pair	298
8.7	Case Study: Sudoku	300
8.8	Multidimensional Arrays	303
<b>Chapter 9</b>	<b>Objects and Classes</b>	<b>323</b>
9.1	Introduction	324
9.2	Defining Classes for Objects	324
9.3	Example: Defining Classes and Creating Objects	326
9.4	Constructing Objects Using Constructors	331
9.5	Accessing Objects via Reference Variables	332
9.6	Using Classes from the Java Library	336
9.7	Static Variables, Constants, and Methods	339
9.8	Visibility Modifiers	344
9.9	Data Field Encapsulation	346
9.10	Passing Objects to Methods	349
9.11	Array of Objects	353
9.12	Immutable Objects and Classes	355
9.13	The Scope of Variables	357
9.14	The <code>this</code> Reference	358
<b>Chapter 10</b>	<b>Object-Oriented Thinking</b>	<b>367</b>
10.1	Introduction	368
10.2	Class Abstraction and Encapsulation	368
10.3	Thinking in Objects	372
10.4	Class Relationships	375
10.5	Case Study: Designing the Course Class	378
10.6	Case Study: Designing a Class for Stacks	380
10.7	Processing Primitive Data Type Values as Objects	382
10.8	Automatic Conversion between Primitive Types and Wrapper Class Types	386
10.9	The <code>BigInteger</code> and <code>BigDecimal</code> Classes	387
10.10	The <code>String</code> Class	388
10.11	The <code>StringBuilder</code> and <code>StringBuffer</code> Classes	395
<b>Chapter 11</b>	<b>Inheritance and Polymorphism</b>	<b>411</b>
11.1	Introduction	412
11.2	Superclasses and Subclasses	412
11.3	Using the <code>super</code> Keyword	418
11.4	Overriding Methods	421
11.5	Overriding vs. Overloading	422
11.6	The <code>Object</code> Class and Its <code>toString()</code> Method	424
11.7	Polymorphism	425
11.8	Dynamic Binding	425
11.9	Casting Objects and the <code>instanceof</code> Operator	429

11.10	The Object's equals Method	433
11.11	The ArrayList Class	434
11.12	Useful Methods for Lists	440
11.13	Case Study: A Custom Stack Class	441
11.14	The protected Data and Methods	442
11.15	Preventing Extending and Overriding	445

## **Chapter 12** Exception Handling and Text I/O **453**

12.1	Introduction	454
12.2	Exception-Handling Overview	454
12.3	Exception Types	459
12.4	Declaring, Throwing, and Catching Exceptions	462
12.5	The finally Clause	470
12.6	When to Use Exceptions	472
12.7	Rethrowing Exceptions	472
12.8	Chained Exceptions	473
12.9	Defining Custom Exception Classes	474
12.10	The File Class	477
12.11	File Input and Output	480
12.12	Reading Data from the Web	487
12.13	Case Study: Web Crawler	488

## **Chapter 13** Abstract Classes and Interfaces **499**

13.1	Introduction	500
13.2	Abstract Classes	500
13.3	Case Study: The Abstract Number Class	505
13.4	Case Study: Calendar and GregorianCalendar	507
13.5	Interfaces	510
13.6	The Comparable Interface	514
13.7	The Cloneable Interface	518
13.8	Interfaces vs. Abstract Classes	523
13.9	Case Study: The Rational Class	526
13.10	Class-Design Guidelines	531

## **Chapter 14** JavaFX Basics **541**

14.1	Introduction	542
14.2	JavaFX vs Swing and AWT	542
14.3	The Basic Structure of a JavaFX Program	542
14.4	Panes, Groups, UI Controls, and Shapes	545
14.5	Property Binding	548
14.6	Common Properties and Methods for Nodes	551
14.7	The Color Class	553
14.8	The Font Class	554
14.9	The Image and ImageView Classes	556
14.10	Layout Panes and Groups	558
14.11	Shapes	567
14.12	Case Study: The ClockPane Class	580

## **Chapter 15** Event-Driven Programming and Animations **593**

15.1	Introduction	594
15.2	Events and Event Sources	596
15.3	Registering Handlers and Handling Events	597
15.4	Inner Classes	601

15.5	Anonymous Inner Class Handlers	602
15.6	Simplifying Event Handling Using Lambda Expressions	605
15.7	Case Study: Loan Calculator	609
15.8	Mouse Events	611
15.9	Key Events	613
15.10	Listeners for Observable Objects	616
15.11	Animation	618
15.12	Case Study: Bouncing Ball	626
15.13	Case Study: US Map	630
<b>Chapter 16</b>	<b>JavaFX UI Controls and Multimedia</b>	<b>643</b>
16.1	Introduction	644
16.2	Labeled and Label	644
16.3	Button	646
16.4	CheckBox	648
16.5	RadioButton	651
16.6	TextField	654
16.7	TextArea	655
16.8	ComboBox	659
16.9	ListView	662
16.10	ScrollBar	665
16.11	Slider	668
16.12	Case Study: Developing a Tic-Tac-Toe Game	671
16.13	Video and Audio	676
16.14	Case Study: National Flags and Anthems	679
<b>Chapter 17</b>	<b>Binary I/O</b>	<b>691</b>
17.1	Introduction	692
17.2	How Is Text I/O Handled in Java?	692
17.3	Text I/O vs. Binary I/O	693
17.4	Binary I/O Classes	694
17.5	Case Study: Copying Files	704
17.6	Object I/O	706
17.7	Random-Access Files	711
<b>Chapter 18</b>	<b>Recursion</b>	<b>719</b>
18.1	Introduction	720
18.2	Case Study: Computing Factorials	720
18.3	Case Study: Computing Fibonacci Numbers	723
18.4	Problem Solving Using Recursion	726
18.5	Recursive Helper Methods	728
18.6	Case Study: Finding the Directory Size	731
18.7	Case Study: Tower of Hanoi	733
18.8	Case Study: Fractals	736
18.9	Recursion vs. Iteration	740
18.10	Tail Recursion	740
<b>Chapter 19</b>	<b>Generics</b>	<b>751</b>
19.1	Introduction	752
19.2	Motivations and Benefits	752
19.3	Defining Generic Classes and Interfaces	754
19.4	Generic Methods	756
19.5	Case Study: Sorting an Array of Objects	758

19.6	Raw Types and Backward Compatibility	760
19.7	Wildcard Generic Types	761
19.8	Erasure and Restrictions on Generics	764
19.9	Case Study: Generic Matrix Class	766
<b>Chapter 20</b>	<b>Lists, Stacks, Queues, and Priority Queues</b>	<b>775</b>
20.1	Introduction	776
20.2	Collections	776
20.3	Iterators	780
20.4	Using the <code>forEach</code> Method	782
20.5	Lists	783
20.6	The <code>Comparator</code> Interface	787
20.7	Static Methods for Lists and Collections	792
20.8	Case Study: Bouncing Balls	795
20.9	Vector and Stack Classes	798
20.10	Queues and Priority Queues	800
20.11	Case Study: Evaluating Expressions	803
<b>Chapter 21</b>	<b>Sets and Maps</b>	<b>815</b>
21.1	Introduction	816
21.2	Sets	816
21.3	Comparing the Performance of Sets and Lists	824
21.4	Case Study: Counting Keywords	827
21.5	Maps	828
21.6	Case Study: Occurrences of Words	833
21.7	Singleton and Unmodifiable Collections and Maps	835
<b>Chapter 22</b>	<b>Developing Efficient Algorithms</b>	<b>839</b>
22.1	Introduction	840
22.2	Measuring Algorithm Efficiency Using Big $O$ Notation	840
22.3	Examples: Determining Big $O$	842
22.4	Analyzing Algorithm Time Complexity	846
22.5	Finding Fibonacci Numbers Using Dynamic Programming	849
22.6	Finding Greatest Common Divisors Using Euclid's Algorithm	851
22.7	Efficient Algorithms for Finding Prime Numbers	855
22.8	Finding the Closest Pair of Points Using Divide-and-Conquer	861
22.9	Solving the Eight Queens Problem Using Backtracking	864
22.10	Computational Geometry: Finding a Convex Hull	867
22.11	String Matching	869
<b>Chapter 23</b>	<b>Sorting</b>	<b>887</b>
23.1	Introduction	888
23.2	Insertion Sort	888
23.3	Bubble Sort	890
23.4	Merge Sort	892
23.5	Quick Sort	896
23.6	Heap Sort	900
23.7	Bucket and Radix Sorts	907
23.8	External Sort	909



<b>Chapter 24</b>	<b>Implementing Lists, Stacks, Queues, and Priority Queues</b>	<b>923</b>
24.1	Introduction	924
24.2	Common Operations for Lists	924
24.3	Array Lists	928
24.4	Linked Lists	935
24.5	Stacks and Queues	949
24.6	Priority Queues	953
<b>Chapter 25</b>	<b>Binary Search Trees</b>	<b>959</b>
25.1	Introduction	960
25.2	Binary Search Trees Basics	960
25.3	Representing Binary Search Trees	961
25.4	Searching for an Element	962
25.5	Inserting an Element into a BST	962
25.6	Tree Traversal	963
25.7	The BST Class	965
25.8	Deleting Elements from a BST	974
25.9	Tree Visualization and MVC	980
25.10	Iterators	983
25.11	Case Study: Data Compression	985
<b>Chapter 26</b>	<b>AVL Trees</b>	<b>995</b>
26.1	Introduction	996
26.2	Rebalancing Trees	996
26.3	Designing Classes for AVL Trees	999
26.4	Overriding the insert Method	1000
26.5	Implementing Rotations	1001
26.6	Implementing the delete Method	1002
26.7	The AVLTree Class	1002
26.8	Testing the AVLTree Class	1008
26.9	AVL Tree Time Complexity Analysis	1011
<b>Chapter 27</b>	<b>Hashing</b>	<b>1015</b>
27.1	Introduction	1016
27.2	What Is Hashing?	1016
27.3	Hash Functions and Hash Codes	1017
27.4	Handling Collisions Using Open Addressing	1019
27.5	Handling Collisions Using Separate Chaining	1023
27.6	Load Factor and Rehashing	1025
27.7	Implementing a Map Using Hashing	1025
27.8	Implementing Set Using Hashing	1034
<b>Chapter 28</b>	<b>Graphs and Applications</b>	<b>1045</b>
28.1	Introduction	1046
28.2	Basic Graph Terminologies	1047
28.3	Representing Graphs	1048
28.4	Modeling Graphs	1054
28.5	Graph Visualization	1064
28.6	Graph Traversals	1067

28.7	Depth-First Search (DFS)	1068
28.8	Case Study: The Connected Circles Problem	1072
28.9	Breadth-First Search (BFS)	1074
28.10	Case Study: The Nine Tails Problem	1077

## Chapter 29 Weighted Graphs and Applications 1091

29.1	Introduction	1092
29.2	Representing Weighted Graphs	1093
29.3	The <code>WeightedGraph</code> Class	1095
29.4	Minimum Spanning Trees	1103
29.5	Finding Shortest Paths	1109
29.6	Case Study: The Weighted Nine Tails Problem	1118

## Chapter 30 Aggregate Operations for Collection Streams 1129

30.1	Introduction	1130
30.2	Stream Pipelines	1130
30.3	<code>IntStream</code> , <code>LongStream</code> , and <code>DoubleStream</code>	1136
30.4	Parallel Streams	1139
30.5	Stream Reduction Using the <code>reduce</code> Method	1141
30.6	Stream Reduction Using the <code>collect</code> Method	1144
30.7	Grouping Elements Using the <code>groupingby</code> Collector	1147
30.8	Case Studies	1150

Chapter 31–44 are available from the Companion Website at [www.pearsonhighered.com/liang](http://www.pearsonhighered.com/liang)

**Chapter 31** Advanced JavaFX and FXML

**Chapter 32** Multithreading and Parallel Programming

**Chapter 33** Networking

**Chapter 34** Java Database Programming

**Chapter 35** Advanced Database Programming

**Chapter 36** Internationalization

**Chapter 37** Servlets

**Chapter 38** JavaServer Pages

**Chapter 39** JavaServer Faces

**Chapter 40** RMI

**Chapter 41** Web Services

<b>Chapter 42</b>	2-4 Trees and B-Trees
<b>Chapter 43</b>	Red-Black Trees
<b>Chapter 44</b>	Testing Using JUnit

<b>APPENDIXES</b>	1161
<b>Appendix A</b>	Java Keywords and Reserved Words 1163
<b>Appendix B</b>	The ASCII Character Set 1164
<b>Appendix C</b>	Operator Precedence Chart 1166
<b>Appendix D</b>	Java Modifiers 1168
<b>Appendix E</b>	Special Floating-Point Values 1170
<b>Appendix F</b>	Number Systems 1171
<b>Appendix G</b>	Bitwise Operations 1175
<b>Appendix H</b>	Regular Expressions 1176
<b>Appendix I</b>	Enumerated Types 1182
<b>Appendix J</b>	The Big-O, Big-Omega, and Big-Theta Notations 1187
<b>QUICK REFERENCE</b>	1189
<b>INDEX</b>	1191