

# TABLE OF CONTENTS

---

iii AP\* Computer Science Review Companion for Fundamentals of Java  
v Preface

## UNIT 1 GETTING STARTED WITH JAVA

<b>3</b>	<b>Chapter 1: Background</b>	74	3.3 Terminal I/O for Different Data Types
3	1.1 History of Computers	77	3.4 Comments
5	1.2 Computer Hardware and Software	82	3.5 Programming Errors
10	1.3 Binary Representation of Information and Computer Memory	87	3.6 Debugging
19	1.4 Programming Languages	92	3.7 Graphics and GUIs: Drawing Shapes and Text
20	1.5 The Software Development Process	100	Summary
23	1.6 Basic Concepts of Object-Oriented Programming		
28	Summary		
<b>31</b>	<b>Chapter 2: First Java Programs</b>	<b>105</b>	<b>Chapter 4: Introduction to Control Statements</b>
31	2.1 Why Java?	105	4.1 Additional Operators
32	2.2 The Java Virtual Machine and Byte Code	107	4.2 Standard Classes and Methods
33	2.3 Choosing a User Interface Style	109	4.3 A Visit to the Farm
34	2.4 Hello World	111	4.4 The <code>if</code> and <code>if-else</code> Statements
37	2.5 Edit, Compile, and Execute	116	4.5 The <code>while</code> Statement
42	2.6 Temperature Conversion	120	4.6 The <code>for</code> Statement
46	2.7 Graphics and GUIs: Windows and Panels	124	4.7 Nested Control Statements and the <code>break</code> Statement
52	Summary	127	4.8 Using Loops with Text Files
<b>57</b>	<b>Chapter 3: Syntax, Errors, and Debugging</b>	133	4.9 Errors in Loops
57	3.1 Language Elements	137	4.10 Graphics and GUIs: I/O Dialog Boxes and Loops
59	3.2 Basic Java Syntax and Semantics	143	Design, Testing, and Debugging Hints
		144	Summary
		149	Unit Review



## **UNIT 2** THE NEXT STEP WITH JAVA

---

<b>155</b>	<b>Chapter 5: Introduction to Defining Classes</b>	<b>247</b>	<b>Chapter 7: Improving the User Interface</b>
155	5.1 The Internal Structure of Classes and Objects	247	7.1 A Thermometer Class
157	5.2 A <code>Student</code> Class	248	7.2 Repeating Sets of Inputs
168	5.3 Editing, Compiling, and Testing the <code>Student</code> Class	249	7.3 A Menu-Driven Conversion Program
174	5.4 The Structure and Behavior of Methods	252	7.4 Formatted Output with <code>printf</code> and <code>format</code>
177	5.5 Scope and Lifetime of Variables	257	7.5 Handling Number Format Exceptions during Input
183	5.6 Graphics and GUIs: Images, a <code>Circle</code> Class, and Mouse Events	258	7.6 Graphics and GUIs
194	Summary	265	Summary
<b>199</b>	<b>Chapter 6: Control Statements Continued</b>	<b>269</b>	<b>Chapter 8: Introduction to HTML and Applets</b>
199	6.1 Logical Operators	269	8.1 Hypertext, Hypermedia, and the World Wide Web
212	6.2 Testing <code>if</code> Statements	271	8.2 Overview of the Hypertext Markup Language
215	6.3 Nested <code>if</code> Statements	275	8.3 Simple Text Elements
218	6.4 Logical Errors in Nested <code>if</code> Statements	277	8.4 Character-Level Formatting
222	6.5 Nested Loops	278	8.5 Lists
224	6.6 Testing Loops	281	8.6 Linking to Other Documents
228	6.7 Loop Verification	283	8.7 Multimedia
234	6.8 Graphics and GUIs: Timers and Animations	286	8.8 Tables
241	Design, Testing, and Debugging Hints	288	8.9 Applets
242	Summary	295	Summary
		299	Unit Review

## UNIT 3 ARRAYS, RECURSION, AND COMPLEXITY

<b>305</b>	<b>Chapter 9: Introduction to Arrays</b>	385	10.10 Exceptions
306	9.1 Conceptual Overview	392	10.11 Reference Types, Equality, and Object Identity
307	9.2 Simple Array Manipulations	395	10.12 Graphics and GUIs: Drawing Multiple Shapes
309	9.3 Looping Through Arrays	402	Summary
311	9.4 Declaring Arrays		
314	9.5 Working with Arrays That Are Not Full		
317	9.6 Parallel Arrays	<b>405</b>	<b>Chapter 11: Arrays Continued</b>
318	9.7 Two-Dimensional Arrays	405	11.1 Advanced Operations on Strings
321	9.8 Using the Enhanced for Loop	411	11.2 Searching
323	9.9 Arrays and Methods	415	11.3 Sorting
326	9.10 Arrays of Objects	422	11.4 Insertions and Removals
338	9.11 Graphics and GUIs: Changing the View of Student Test Scores	426	11.5 Working with Arrays of Objects
343	Design, Testing, and Debugging Hints	429	11.6 The Class <code>java.util.ArrayList</code>
343	Summary	442	11.7 Graphics and GUIs: Menus
		450	Summary
<b>351</b>	<b>Chapter 10: Classes Continued</b>	<b>455</b>	<b>Chapter 12: Recursion, Complexity, and Searching and Sorting</b>
352	10.1 Class ( <code>static</code> ) Variables and Methods	456	12.1 Recursion
356	10.2 <code>TurtleGraphics</code>	468	12.2 Complexity Analysis
359	10.3 Java Interfaces—The Client Perspective	475	12.3 Binary Search
362	10.4 Java Interfaces—The Implementation Perspective	478	12.4 Quicksort
367	10.5 Code Reuse Through Inheritance	482	12.5 Merge Sort
372	10.6 Inheritance and Abstract Classes	489	12.6 Graphics and GUIs: Drawing Recursive Patterns
377	10.7 Some Observations about Interfaces, Inheritance, and Relationships among Classes	497	Design, Testing, and Debugging Hints
380	10.8 Acceptable Classes for Parameters and Return Values	497	Summary
383	10.9 Error Handling with Classes	501	Unit Review

## UNIT 4 USING ABSTRACT DATA TYPES

<b>507</b>	<b>Chapter 13: Object-Oriented Analysis and Design</b>	544	13.8 Graphics and GUIs: Text Areas, Scrolling List Boxes, and a GUI-Based Adventure Game
508	13.1 Introduction	553	Summary
509	13.2 Overview of Analysis and Design		
510	13.3 Request	<b>557</b>	<b>Chapter 14: Linear Collections: Lists</b>
511	13.4 Analysis	558	14.1 Overview of Collections
522	13.5 Interlude: Support Classes for Menu-Driven Programs	562	14.2 Lists
528	13.6 Design	572	14.3 Iterators
536	13.7 Implementation	578	Summary

<b>583</b>	<b>Chapter 15: Linear Collections: Stacks and Queues</b>	<b>625</b>	<b>Chapter 16: Unordered Collections: Sets and Maps</b>
583	15.1 Stacks	626	16.1 Sets
596	15.2 The <code>StringTokenizer</code> Class	629	16.2 Sorted Sets
601	15.3 Queues and Priority Queues	632	16.3 Example Application: Word Frequencies
617	15.4 Graphics and GUIs: Check Boxes and Radio Buttons	634	16.4 Maps
620	Summary	638	16.5 Sorted Maps
		644	16.6 The Glue that Holds Collections Together
		649	Summary
		653	Unit Review



## UNIT 5 IMPLEMENTING ABSTRACT DATA TYPES

<b>659</b>	<b>Chapter 17: Implementing Lists</b>	699	18.3 Overview of Hashing
659	17.1 Interfaces, Multiple Implementations, and Prototypes	704	18.4 Hashing Implementation of Maps
661	17.2 The <code>IndexedList</code> Interface	709	18.5 Hashing Implementation of Sets
663	17.3 The Fixed-Size Array Implementation of Indexed Lists	712	Summary
664	17.4 The Singly Linked Implementation of Indexed Lists	<b>715</b>	<b>Chapter 19: Implementing Trees and Priority Queues</b>
674	17.5 Complexity Analysis of Indexed List Implementations	715	19.1 An Overview of Trees
677	17.6 Positional Lists	724	19.2 Binary Tree Traversals
682	17.7 Fixed-Size Array Implementation of Positional Lists	727	19.3 A Linked Implementation of Binary Trees
685	17.8 Doubly Linked Implementation of Positional Lists	737	19.4 An Array Implementation of Binary Trees
690	17.9 Complexity Analysis of Positional Lists	740	19.5 Implementing Heaps
691	17.10 Iterators	743	19.6 Using a Heap to Implement a Priority Queue
691	Summary	744	Summary
		749	Unit Review
<b>695</b>	<b>Chapter 18: Implementing Sets and Maps</b>		
696	18.1 The Set and Map Prototypes		
698	18.2 List Implementations of Sets and Maps		

<b>Appendix A-1</b>	<b>Appendix A</b> Java Resources
<b>Appendix B-1</b>	<b>Appendix B</b> Java Language Elements
<b>Appendix C-1</b>	<b>Appendix C</b> Operator Precedence
<b>Appendix D-1</b>	<b>Appendix D</b> ASCII Character Set
<b>Appendix E-1</b>	<b>Appendix E</b> Number Systems
<b>Appendix F-1</b>	<b>Appendix F</b> Java Exception Handling
<b>Appendix G-1</b>	<b>Appendix G</b> Java Packages and jar Files
<b>Appendix H-1</b>	<b>Appendix H</b> Files
<b>Appendix I-1</b>	<b>Appendix I</b> The <code>TurtleGraphics</code> Package
<b>Appendix J-1</b>	<b>Appendix J</b> AP Correlations
<b>Glossary-1</b>	<b>Glossary</b>
<b>Index-1</b>	<b>Index</b>