

INTRODUCTION to

java™ and SOFTWARE Design

Nell Dale

Chip Weems

Mark Headington

JONES AND BARTLETT COMPUTER SCIENCE



Preface v

1 Overview of Programming and Problem Solving 1

1.1 Overview of Programming 2

What Is Programming? 2

How Do We Write a Program? 3

Theoretical Foundations: Binary Representation of Data 8

1.2 How Is a Program Converted into a Form That a Computer Can Use? 9

1.3 How is Interpreting a Program Different from Executing It? 13

1.4 How Is Compilation Related to Interpretation and Execution? 14

1.5 What Kinds of Instructions Can be Written in a Programming Language? 16

Object-Oriented Programming Languages 18

1.6 What Is a Computer? 19

1.7 Ethics and Responsibilities in the Computing Profession 22

Software Piracy 24

Privacy of Data 24

Use of Computer Resources 25

Software Engineering 26

1.8	Problem-Solving Techniques	27
	Ask Questions	27
	Look for Things That Are Familiar	28
	Solve by Analogy	28
	Means-Ends Analysis	29
	Divide and Conquer	30
	The Building-Block Approach	31
	Merging Solutions	31
	Mental Blocks: The Fear of Starting	32
	Algorithmic Problem Solving	33
	Problem-Solving Case Study: A Company Payroll Program	33
	Summary	40
	Quick Check	41
	Exam Preparation Exercises	42
	Programming Warm-Up Exercises	44
	Case Study Follow-Up Exercises	44

2

Java Syntax and Semantics, and the Program Entry Process 47

2.1	Syntax and Semantics	48
	<i>Theoretical Foundations: Metalanguages</i>	49
	Syntax Templates	51
	Naming Program Elements: Identifiers	52
	<i>Matters of Style: Using Meaningful, Readable Identifiers</i>	54
2.2	Data Types	54
	<i>Background Information: Data Storage</i>	55
2.3	Classes and Objects	56
2.4	Defining Terms: Declarations	59
	<i>Matters of Style: Capitalization of Identifiers</i>	63
2.5	Taking Action: Executable Statements	64
	Beyond Minimalism: Adding Comments to a Program	70
	Program Construction	71
	Blocks	73
2.6	Program Entry, Correction, and Execution	75
	Entering a Program	75
	Compiling and Running a Program	76
	Finishing Up	77

Problem-Solving Case Study: Display a Date in Multiple Formats	78
Testing and Debugging	81
Summary of Classes	82
Summary	83
Quick Check	83
Exam Preparation Exercises	86
Programming Warm-Up Exercises	88
Programming Problems	90
Case Study Follow-Up Exercises	91

3	Event-Driven Output	93
3.1	Classes and Methods	94
	Frames	94
	<i>Background Information: The Origins of Java</i>	101
3.2	Formatting Output	103
	Using <code>GridLayout</code> for Tabular Output	103
	Alignment of Text within Labels	104
3.3	Handling Events	105
3.4	Register an Event Listener	106
3.5	Event-Handler Methods	108
	Problem-Solving Case Study: Display a Date in Multiple Formats	112
	Testing and Debugging	115
	Summary of Classes	116
	Summary	117
	Quick Check	117
	Exam Preparation Exercises	118
	Programming Warm-Up Exercises	119
	Programming Problems	121
	Case Study Follow-Up Exercises	122

4	Numeric Types and Expressions	123
4.1	Overview of Java Data Types	124
4.2	Numeric Data Types	126
	Integral Types	126
	Floating-Point Types	128

4.3	Declarations for Numeric Types	129
	Named Constant Declarations	129
	<i>Software Engineering Tip: Using Named Constants Instead of Literals</i>	130
	Variable Declarations	130
4.4	Simple Arithmetic Expressions	131
	Arithmetic Operators	131
	Increment and Decrement Operators	135
4.5	Compound Arithmetic Expressions	136
	Precedence Rules	136
	Type Conversion and Type Casting	138
	<i>May We Introduce: Blaise Pascal</i>	144
4.6	Additional Mathematical Methods	145
	<i>Matters of Style: Program Formatting</i>	146
4.7	Additional <code>String</code> Operations	149
	The <code>length</code> Method	149
	The <code>substring</code> Method	151
4.8	Formatting Numeric Types	153
	<i>Software Engineering Tip: Understanding Before Changing</i>	154
	Problem-Solving Case Study: Map Measurements	155
	Testing and Debugging	160
	Summary of Classes	161
	Summary	162
	Quick Check	162
	Exam Preparation Exercises	164
	Programming Warm-Up Exercises	168
	Programming Problems	171
	Case Study Follow-Up Exercises	173

5 Event-Driven Input and Software Design Strategies 175

5.1	Getting Data into Programs	176
5.2	Entering Data Using Fields in a Frame	177
5.3	Creating a Data Entry Field	179
5.4	Extracting a Value from a Field	180
5.5	Creating a Button	182
5.6	Creating and Registering a Button Event Listener	184

5.7	Handling a Button Event	188
5.8	Interactive Input/Output	191
5.9	Converting Strings to Numeric Values	192
5.10	Noninteractive Input/Output	195
5.11	Software Design Strategies	195
5.12	What are Objects and Classes?	196
5.13	Object-Oriented Design	200
	Object-Oriented Problem Solving	201
	Identifying the Classes	202
	Initial Responsibilities	204
	A First Scenario Walk-Through	204
	Subsequent Scenarios	207
	Inheritance	208
	Enhancing CRC Cards with Additional Information	208
5.14	Functional Decomposition	210
	Writing Modules	211
	Pseudocode	212
	<i>Software Engineering Tip: Documentation</i>	212
	Problem-Solving Case Study: Averaging Rainfall Amounts	213
	<i>Background Information: Programming at Many Scales</i>	220
	Testing and Debugging	221
	Summary of Classes	223
	Summary	224
	Quick Check	225
	Exam Preparation Exercises	226
	Programming Warm-Up Exercises	227
	Programming Problems	228
	Case Study Follow-Up Exercises	229

6 Conditions, Logical Expressions, and Selection Control Structures 231

6.1	Flow of Control	232
	Selection	233
6.2	Conditions and Logical Expressions	234
	The boolean Data Type	234
	Logical Expressions	234

	<i>Background Information: George Boole</i>	242
	Precedence of Operators	243
	<i>Software Engineering Tip: Changing English Statements into Logical Expressions</i>	245
	Relational Operators with Floating-Point Types	246
6.3	The if statement	247
	The if-else Form	247
	Blocks (Compound Statements)	249
	<i>Matters of Style: Braces and Blocks</i>	251
	The if Form	252
6.4	Nested if Statements	254
	The Dangling else	257
6.5	Handling Multiple Button Events	258
	Problem-Solving Case Study: A Simple Calculator	262
	Testing and Debugging	270
	Summary of Classes	277
	Summary	278
	Quick Check	278
	Exam Preparation Exercises	279
	Programming Warm-Up Exercises	283
	Programming Problems	286
	Case Study Follow-Up Exercises	289

7 Classes and Methods 293

7.1	Encapsulation	294
	Abstraction	296
	<i>Theoretical Foundations: Categories of Instance Responsibility</i>	298
7.2	Class Interface Design	299
	Public Interface Design	300
7.3	Internal Data Representation	302
	Data Lifetime	304
	Internal Representation Example	307
7.4	Class Syntax	307
7.5	Declaring Methods	308
	Parameters	311

	A Parameter-Passing Analogy	315
	Implementing a Responsibility as a Method	316
	Constructors	318
7.6	Packages	320
	Package Syntax	320
	Packages with Multiple Compilation Units	321
	Problem-Solving Case Study: Implementing the Name Class	322
	Testing and Debugging	328
	Summary of Classes	332
	Summary	332
	Quick Check	333
	Exam Preparation Exercises	334
	Programming Warm-Up Exercises	336
	Programming Assignments	336
	Case Study Follow-Up Exercises	337

8 Inheritance, Polymorphism, and Scope 339

8.1	Inheritance	340
	An Analogy	340
8.2	Inheritance and the Object-Oriented Design Process	342
8.3	How to Read a Class Hierarchy	346
	Overriding	350
	Hiding	350
	Polymorphism	350
8.4	Derived Class Syntax	351
	<i>May We Introduce: Ada Lovelace</i>	352
8.5	Scope of Access	353
	Internal Scope	354
	External Scope	357
8.6	Implementing a Derived Class	360
	Constructors in Derived Classes	361
	Overloading and Method Signatures	361
	Accessing Overridden and Hidden Methods and Fields	363
8.7	Copy Constructors	365
	Problem-Solving Case Study: Extending <code>TextField</code> for Numeric Input and Output	367

Testing and Debugging	371
Summary of Classes	378
Summary	378
Quick Check	379
Exam Preparation Exercises	380
Programming Warm-Up Exercises	381
Programming Problems	383
Case Study Follow-Up Exercises	383

9

File I/O and Looping 385

9.1	File Input and Output	386
	Files	386
	Using Files	387
	Extending File I/O with <code>PrintWriter</code> and <code>BufferedReader</code>	393
	Exceptions with Input and Output	398
	An Example Program Using Files	400
9.2	Looping	402
	The <code>while</code> Statement	403
	Phases of Loop Execution	404
	Loops Using the <code>while</code> Statement	405
	Count-Controlled Loops	406
	Event-Controlled Loops	407
	Looping Subtasks	412
	How to Design Loops	415
	Designing the Flow of Control	415
	Designing the Process within the Loop	417
	The Loop Exit	418
	Nested Loops	418
	General Pattern	419
	Designing Nested Loops	422
	<i>Theoretical Foundations: Analysis of Algorithms</i>	424
	Problem-Solving Case Study: Average Income by Gender	429
	Testing and Debugging	435
	Summary of Classes	439
	Summary	440

Quick Check	442
Exam Preparation Exercises	443
Programming Warm-Up Exercises	446
Programming Problems	448
Case Study Follow-Up Exercises	449

10 Additional Control Structures and Exceptions 451

10.1	Additional Control Structures	452
	The <code>switch</code> Statement	452
	<i>May We Introduce: Admiral Grace Murray Hopper</i>	457
	The <code>do</code> Statement	458
	The <code>for</code> Statement	460
	Guidelines for Choosing a Looping Statement	463
10.2	Exception-Handling Mechanism	463
	The <code>try-catch-finally</code> Statement	463
	Generating an Exception with <code>throw</code>	466
	Exception Classes	469
10.3	Additional Java Operators	470
	Assignment Operators and Assignment Expressions	471
	Increment and Decrement Operators	472
	The <code>?:</code> Operator	474
	Operator Precedence	474
	Problem-Solving Case Study: Monthly Rainfall Averages	479
	Testing and Debugging	484
	Summary of Classes	484
	Summary	485
	Quick Check	485
	Exam Preparation Exercises	487
	Programming Warm-Up Exercises	489
	Programming Problems	490
	Case Study Follow-Up Exercises	493

11 One-Dimensional Arrays 495

11.1	Atomic Data Types	496
11.2	Composite Data Types	498
11.3	One-Dimensional Arrays	499

	Accessing Individual Components	505
	Out-of-Bounds Array Indexes	507
	Aggregate Array Operations	508
11.4	Examples of Declaring and Processing Arrays	510
	Occupancy Rates	510
	Sales Figures	512
	Letter Counts	513
11.5	Arrays of Objects	514
	Arrays of Strings	514
	Arrays of User-Defined Objects	517
11.6	Arrays and Methods	518
11.7	Special Kinds of Array Processing	518
	Partial (or Sub) Array Processing	519
	Indexes with Semantic Content	519
	Problem-Solving Case Study: Comparison of Two Lists	519
	Problem-Solving Case Study: Grading True/False Tests	525
	Testing and Debugging	534
	Summary of Classes	536
	Summary	537
	Quick Check	537
	Exam Preparation Exercises	538
	Programming Warm-Up Exercises	541
	Programming Problems	542
	Case Study Follow-Up Exercises	544

12 Array-Based Lists 545

12.1	Lists	546
12.2	List Class	548
	Brainstorming the List Class	548
	CRC Card	548
	Refining the Responsibilities	549
	Internal Data Representation	552
	Responsibility Algorithms for Class List	552
	Test Plan	558
12.3	Sorting the List Items	559
	Responsibility Algorithms for Class ListWithSort	560

	Class ListWithSort	561
12.4	Sorted List	563
	Brainstorming Sorted List	563
	Responsibility Algorithms for Class SortedList	565
	Test Plan	569
12.5	The List Class Hierarchy and Abstract Classes	569
12.6	Searching	571
	Sequential Search	571
	Binary Search	572
	<i>Theoretical Foundations: Complexity of Searching and Sorting</i>	578
12.7	Generic Lists	579
	Comparable Interface	579
	Polymorphism	582
	Problem-Solving Case Study: Exam Attendance	583
	Testing and Debugging	592
	Summary of Classes	592
	Summary	593
	Quick Check	594
	Exam Preparation Exercises	595
	Programming Warm-Up Exercises	596
	Programming Problems	597
	Case Study Follow-Up Exercises	598

13 Multidimensional Arrays and Numeric Computation 599

13.1	Two-Dimensional Arrays	600
	Array Declaration and Instantiation	601
	Accessing Individual Components	602
	Using Initializer Lists	603
13.2	Processing Two-Dimensional Arrays	604
	Sum the Rows	605
	Sum the Columns	607
	Initialize the Array	608
	Row Processing	609
	Column Processing	609
	Two-Dimensional Arrays and Methods	609

13.3	Multidimensional Arrays	610
13.4	Vector Class	611
	Floating-Point Numbers	612
	Representation of Floating-Point Numbers	612
	Arithmetic with Floating-Point Numbers	615
	Implementation of Floating-Point Numbers in the Computer	616
	<i>Background Information: Practical Implications of Limited Precision</i>	621
	<i>Software Engineering Tip: Choosing a Numeric Data Type</i>	622
	Problem-Solving Case Study: Matrix Manipulation	623
	Testing and Debugging	633
	Summary of Classes	634
	Summary	635
	Quick Check	636
	Exam Preparation Exercises	637
	Programming Warm-Up Exercises	640
	Programming Problems	642
	Case Study Follow-Up Exercises	645

14 Recursion 647

14.1	What is Recursion?	648
	Power Function Definition	648
	Power Function Implementation	649
14.2	More Examples with Simple Variables	651
	Calculating the Factorial Function	651
	Iterative Solution	654
	Recursive Solution	654
	Converting Decimal Integers to Binary	654
	Towers of Hanoi	658
14.3	Recursive Algorithms with Structured Variables	662
	Printing the Values in an Array	662
	Binary Search	664
14.4	Recursion or Iteration?	666
	Testing and Debugging	666
	Summary	667
	Quick Check	667

Exam Preparation Exercises	668
Programming Warm-Up Exercises	669
Programming Problems	671
Appendix A	Java Reserved Words 673
Appendix B	Operator Precedence 674
Appendix C	Primitive Data Types 675
Appendix D	ASCII Subset of Unicode 676
Appendix E	Decimal Format Type 678
Appendix F	Program Style, Formatting, and Documentation 683
Appendix G	Applets 688
Glossary	693
Answers to Selected Exercises	709
Index	745