

# DETAILED CONTENTS

## PREFACE

<b>WELCOME – AND THANK YOU!</b> .....	<b>xlvi</b>
<b>TARGET AUDIENCE</b> .....	<b>xlvi</b>
<b>APPROACH(ES)</b> .....	<b>xlvi</b>
<b>ORGANIZATION</b> .....	<b>xlvi</b>
<i>PART I: THE JAVA STUDENT SURVIVAL GUIDE</i> .....	<i>xlvi</i>
CHAPTER 1 - AN APPROACH TO THE ART OF PROGRAMMING .....	<i>xlvi</i>
CHAPTER 2 - SMALL VICTORIES.....	<i>xlvi</i>
CHAPTER 3 - PROJECT WALKTHROUGH: A COMPLETE EXAMPLE .....	<i>xlvii</i>
CHAPTER 4 - COMPUTERS, PROGRAMS, AND ALGORITHMS .....	<i>xlvii</i>
<i>PART II: LANGUAGE FUNDAMENTALS</i> .....	<i>xlvii</i>
CHAPTER 5 - OVERVIEW OF THE JAVA API.....	<i>xlvii</i>
CHAPTER 6 - SIMPLE JAVA PROGRAMS: USING PRIMITIVE AND REFERENCE DATA TYPES.....	<i>xlvii</i>
CHAPTER 7 - CONTROLLING THE FLOW OF PROGRAM EXECUTION .....	<i>xlvii</i>
CHAPTER 8 - ARRAYS.....	<i>xlvii</i>
CHAPTER 9 - TOWARD PROBLEM ABSTRACTION: CREATING NEW DATA TYPES .....	<i>xlviii</i>
CHAPTER 10 - COMPOSITIONAL DESIGN .....	<i>xlviii</i>
CHAPTER 11 - EXTENDING CLASS BEHAVIOR THROUGH INHERITANCE .....	<i>xlviii</i>
<i>PART III: GRAPHICAL USER INTERFACE PROGRAMMING</i> .....	<i>xlviii</i>
CHAPTER 12 - JAVA SWING API OVERVIEW.....	<i>xlviii</i>
CHAPTER 13 - HANDLING GUI EVENTS.....	<i>xlviii</i>
CHAPTER 14 - AN ADVANCED GUI PROJECT .....	<i>xlviii</i>
<i>PART IV: INTERMEDIATE CONCEPTS</i> .....	<i>xlxi</i>
CHAPTER 15 - EXCEPTIONS.....	<i>xlxi</i>
CHAPTER 16 - THREADS.....	<i>xlxi</i>
CHAPTER 17 - COLLECTIONS.....	<i>xlxi</i>
CHAPTER 18 - FILE I/O.....	<i>xlxi</i>
<i>PART V: NETWORK PROGRAMMING</i> .....	<i>xlxi</i>
CHAPTER 19 - INTRODUCTION TO NETWORKING AND DISTRIBUTED APPLICATIONS .....	<i>l</i>
CHAPTER 20 - CLIENT-SERVER APPLICATIONS.....	<i>l</i>
CHAPTER 21 - APPLETS AND JDBC.....	<i>l</i>
<i>PART VI: OBJECT-ORIENTED PROGRAMMING</i> .....	<i>l</i>
CHAPTER 22 - INHERITANCE, COMPOSITION, INTERFACES, AND POLYMORPHISM.....	<i>l</i>
CHAPTER 23 - WELL-BEHAVED OBJECTS.....	<i>li</i>
CHAPTER 24 - THREE DESIGN PRINCIPLES.....	<i>li</i>
CHAPTER 25 - HELPFUL DESIGN PATTERNS .....	<i>li</i>
<b>PEDAGOGY</b> .....	<b>li</b>
LEARNING OBJECTIVES.....	<i>li</i>
INTRODUCTION.....	<i>li</i>
CONTENT.....	<i>li</i>
QUICK REVIEWS.....	<i>li</i>
SUMMARY .....	<i>lii</i>
SKILL-BUILDING EXERCISES .....	<i>lii</i>
SUGGESTED PROJECTS.....	<i>lii</i>
SELF-TEST QUESTIONS .....	<i>lii</i>
REFERENCES.....	<i>lii</i>
NOTES .....	<i>lii</i>
<b>TYPOGRAPHICAL FORMATS</b> .....	<b>lii</b>
<i>THIS IS AN EXAMPLE OF A FIRST-LEVEL SUBHEADING</i> .....	<i>lii</i>

<i>This Is An Example Of A Second-Level Subheading</i> .....	<i>lii</i>
<i>SOURCE CODE FORMATTING</i> .....	<i>liii</i>
<b>CD-ROM</b> .....	<b>liii</b>
<b>SUPPORTSITE™ WEBSITE</b> .....	<b>liii</b>
<b>PROBLEM REPORTING</b> .....	<b>liii</b>
<b>ABOUT THE AUTHORS</b> .....	<b>liii</b>
<b>ACKNOWLEDGMENTS</b> .....	<b>liv</b>

# PART I: THE JAVA STUDENT SURVIVAL GUIDE

## I AN APPROACH TO THE ART OF PROGRAMMING

<b>INTRODUCTION</b> .....	<b>4</b>
<i>THE DIFFICULTIES YOU WILL ENCOUNTER LEARNING JAVA</i> .....	<i>4</i>
<i>REQUIRED SKILLS</i> .....	<i>4</i>
<i>THE PLANETS WILL COME INTO ALIGNMENT</i> .....	<i>4</i>
<i>HOW THIS CHAPTER WILL HELP YOU</i> .....	<i>5</i>
<b>PROJECT MANAGEMENT</b> .....	<b>5</b>
<i>THREE SOFTWARE DEVELOPMENT ROLES</i> .....	<i>5</i>
<i>ANALYST</i> .....	<i>5</i>
<i>ARCHITECT</i> .....	<i>5</i>
<i>PROGRAMMER</i> .....	<i>6</i>
<i>A PROJECT-APPROACH STRATEGY</i> .....	<i>6</i>
<i>YOU HAVE BEEN HANDED A PROJECT – NOW WHAT?</i> .....	<i>6</i>
<i>STRATEGY AREAS OF CONCERN</i> .....	<i>6</i>
<i>THINK ABSTRACTLY</i> .....	<i>8</i>
<i>THE STRATEGY IN A NUTSHELL</i> .....	<i>8</i>
<i>APPLICABILITY TO THE REAL WORLD</i> .....	<i>8</i>
<b>THE ART OF PROGRAMMING</b> .....	<b>8</b>
<i>DON'T START AT THE COMPUTER</i> .....	<i>9</i>
<i>INSPIRATION STRIKES AT THE WEIRDEST TIME</i> .....	<i>9</i>
<i>OWN YOUR OWN COMPUTER</i> .....	<i>9</i>
<i>YOU EITHER HAVE TIME AND NO MONEY, OR MONEY AND NO TIME</i> .....	<i>9</i>
<i>THE FAMILY COMPUTER IS NOT GOING TO CUT IT!</i> .....	<i>9</i>
<i>SET THE MOOD</i> .....	<i>10</i>
<i>LOCATION, LOCATION, LOCATION</i> .....	<i>10</i>
<i>CONCEPT OF THE FLOW</i> .....	<i>10</i>
<i>THE STAGES OF FLOW</i> .....	<i>10</i>
<i>BE EXTREME</i> .....	<i>11</i>
<i>THE PROGRAMMING CYCLE</i> .....	<i>11</i>
<i>THE PROGRAMMING CYCLE SUMMARIZED</i> .....	<i>12</i>
<i>A HELPFUL TRICK: STUBBING</i> .....	<i>12</i>
<i>FIX THE FIRST COMPILER ERROR FIRST</i> .....	<i>12</i>
<b>MANAGING PROJECT COMPLEXITY</b> .....	<b>12</b>
<i>CONCEPTUAL COMPLEXITY</i> .....	<i>12</i>
<i>MANAGING CONCEPTUAL COMPLEXITY</i> .....	<i>13</i>
<i>PHYSICAL COMPLEXITY</i> .....	<i>13</i>
<i>THE RELATIONSHIP BETWEEN PHYSICAL AND CONCEPTUAL COMPLEXITY</i> .....	<i>14</i>
<i>MANAGING PHYSICAL COMPLEXITY</i> .....	<i>14</i>
<i>MAXIMIZE COHESION – MINIMIZE COUPLING</i> .....	<i>14</i>
<b>JAVA SOURCE FILE STRUCTURE</b> .....	<b>14</b>
<i>SAMPLECLASS IN ACTION</i> .....	<i>15</i>
<i>GENERAL RULES FOR CREATING JAVA SOURCE FILES</i> .....	<i>16</i>

<i>Rule-of-Thumb: ONE CLASS PER FILE</i> .....	17
<i>AVOID ANONYMOUS, NESTED, AND INNER CLASSES</i> .....	17
<i>CREATE A SEPARATE MAIN APPLICATION FILE</i> .....	17
<b>PACKAGES</b> .....	<b>17</b>
<b>COMMENTING</b> .....	<b>18</b>
<i>SINGLE-LINE COMMENTS</i> .....	18
<i>MULTI-LINE COMMENTS</i> .....	18
<i>JAVADOC COMMENTS</i> .....	19
<i>GENERATING JAVADOC EXAMPLE OUTPUT</i> .....	19
<b>IDENTIFIER NAMING - WRITING SELF-COMMENTING CODE</b> .....	<b>19</b>
<i>BENEFITS OF SELF-COMMENTING CODE</i> .....	20
<i>CODING CONVENTION</i> .....	20
<i>CLASS NAMES</i> .....	21
<i>CONSTANT NAMES</i> .....	21
<i>VARIABLE NAMES</i> .....	21
<i>METHOD NAMES</i> .....	22
<b>SUMMARY</b> .....	<b>22</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>22</b>
<b>SUGGESTED PROJECTS</b> .....	<b>24</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>24</b>
<b>REFERENCES</b> .....	<b>25</b>
<b>NOTES</b> .....	<b>25</b>

## 2 SMALL VICTORIES: CREATING JAVA PROJECTS

<b>INTRODUCTION</b> .....	<b>28</b>
<b>JAVA PROJECT CREATION PROCESS</b> .....	<b>28</b>
<i>SOURCE FILE CREATION STAGE</i> .....	29
<i>SOURCE FILE COMPILATION STAGE</i> .....	29
<i>WHAT IF A SOURCE FILE CONTAINS ERRORS?</i> .....	29
<i>MAIN APPLICATION EXECUTION STAGE</i> .....	29
<i>AUTOMATING THE JAVA PROJECT CREATION PROCESS</i> .....	30
<b>INTEGRATED DEVELOPMENT ENVIRONMENTS</b> .....	<b>30</b>
<i>TEXTPAD™ - J2SDK COMBINATION</i> .....	30
<i>SUN'S NETBEANS™</i> .....	30
<i>BORLAND® JBUILDER®</i> .....	30
<i>ECLIPSE®</i> .....	30
<b>CREATING JAVA PROJECTS USING MICROSOFT® WINDOWS® 2000/XP</b> .....	<b>31</b>
<i>SUN'S JAVA 2 STANDARD EDITION SOFTWARE DEVELOPMENT KIT (J2SDK)</i> .....	31
<i>STEP 1: DOWNLOAD AND INSTALL THE J2SDK</i> .....	31
<i>STEPS 2 &amp; 3: SETTING MICROSOFT WINDOWS ENVIRONMENT VARIABLES</i> .....	31
<i>ANOTHER HELPFUL HINT: DISPLAY FILES SUFFIXES AND FULL PATH NAMES</i> .....	36
<i>STEPS 4 &amp; 5: SELECT A TEXT EDITOR AND CREATE THE SOURCE FILES</i> .....	36
<i>STEP 6: COMPILING THE SOURCE FILES</i> .....	38
<i>STEP 7: EXECUTING THE MAIN APPLICATION CLASS</i> .....	38
<i>BORLAND'S JBUILDER</i> .....	39
<i>STEPS TO CREATING JAVA PROJECTS IN JBUILDER</i> .....	39
<i>QUICK REVIEW</i> .....	45
<b>CREATING JAVA PROJECTS USING LINUX™</b> .....	<b>45</b>
<i>A BRIEF NOTE ON THE MANY FLAVORS OF LINUX</i> .....	46
<i>OBTAINING AND INSTALLING SUN'S J2SDK FOR LINUX</i> .....	46
<i>SELECTING AN INSTALLATION LOCATION</i> .....	47
<i>SETTING THE PATH ENVIRONMENT VARIABLE (BASH SHELL)</i> .....	48
<i>SETTING THE CLASSPATH ENVIRONMENT VARIABLE (BASH)</i> .....	49
<i>CREATING, COMPILING, AND RUNNING JAVA PROGRAMS</i> .....	49
<i>CREATING JAVA SOURCE FILES</i> .....	49

<i>Compiling JAVA SOURCE Files</i> .....	50
<i>Running The Application</i> .....	50
<i>Quick Review</i> .....	51
<b>CREATING JAVA PROJECTS USING MACINTOSH OS X DEVELOPER TOOLS</b> .....	<b>51</b>
<i>CREATING A JAVA PROJECT WITH XCODE</i> .....	51
<i>Quick Review</i> .....	52
<b>CREATING AND RUNNING EXECUTABLE JAR FILES</b> .....	<b>53</b>
<i>STEPS TO CREATE AN EXECUTABLE JAR FILE</i> .....	53
<i>STEP 1: CREATE YOUR JAVA PROGRAM</i> .....	53
<i>STEP 2: CREATE MANIFEST FILE</i> .....	54
<i>STEP 3: USE THE jar COMMAND</i> .....	54
<i>EXECUTING THE jar FILE</i> .....	54
<i>Quick Review</i> .....	54
<b>SUMMARY</b> .....	<b>55</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>55</b>
<b>SUGGESTED PROJECTS</b> .....	<b>56</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>56</b>
<b>REFERENCES</b> .....	<b>57</b>
<b>NOTES</b> .....	<b>57</b>

## 3 PROJECT WALKTHROUGH: A COMPLETE EXAMPLE

<b>INTRODUCTION</b> .....	<b>60</b>
<b>THE PROJECT-APPROACH STRATEGY SUMMARIZED</b> .....	<b>60</b>
<b>DEVELOPMENT CYCLE</b> .....	<b>61</b>
<b>PROJECT SPECIFICATION</b> .....	<b>62</b>
<i>ANALYZING THE PROJECT SPECIFICATION</i> .....	63
<i>APPLICATION REQUIREMENTS STRATEGY AREA</i> .....	63
<i>PROBLEM DOMAIN STRATEGY AREA</i> .....	64
<i>LANGUAGE FEATURES STRATEGY AREA</i> .....	66
<i>DESIGN STRATEGY AREA</i> .....	68
<b>DEVELOPMENT CYCLE FIRST ITERATION</b> .....	<b>69</b>
<i>PLAN (FIRST ITERATION)</i> .....	69
<i>CODE (FIRST ITERATION)</i> .....	70
<i>TEST (FIRST ITERATION)</i> .....	70
<i>INTEGRATE/TEST (FIRST ITERATION)</i> .....	70
<b>DEVELOPMENT CYCLE SECOND ITERATION</b> .....	<b>70</b>
<i>PLAN (SECOND ITERATION)</i> .....	71
<i>CODE (SECOND ITERATION)</i> .....	71
<i>TEST (SECOND ITERATION)</i> .....	71
<i>INTEGRATE/TEST (SECOND ITERATION)</i> .....	72
<b>DEVELOPMENT CYCLE THIRD ITERATION</b> .....	<b>72</b>
<i>PLAN (THIRD ITERATION)</i> .....	72
<i>CODE (THIRD ITERATION)</i> .....	73
<i>TEST (THIRD ITERATION)</i> .....	75
<i>INTEGRATE/TEST (THIRD ITERATION)</i> .....	75
<b>DEVELOPMENT CYCLE FOURTH ITERATION</b> .....	<b>75</b>
<i>PLAN (FOURTH ITERATION)</i> .....	75
<i>IMPLEMENTING STATE TRANSITION DIAGRAMS</i> .....	76
<i>IMPLEMENTING THE printFloor() METHOD</i> .....	77
<i>CODE (FOURTH ITERATION)</i> .....	77
<i>TEST (FOURTH ITERATION)</i> .....	79
<i>INTEGRATE/TEST (FOURTH ITERATION)</i> .....	79
<b>DEVELOPMENT CYCLE FIFTH ITERATION</b> .....	<b>79</b>
<i>PLAN (FIFTH ITERATION)</i> .....	80
<i>CODE (FIFTH ITERATION)</i> .....	81

<i>TEST (Fifth ITERATION)</i> .....	82
<i>INTEGRATE/TEST (Fifth ITERATION)</i> .....	82
<b>SOME FINAL CONSIDERATIONS</b> .....	<b>82</b>
<b>COMPLETE ROBOTRAT.JAVA SOURCE CODE LISTING</b> .....	<b>84</b>
<b>SUMMARY</b> .....	<b>88</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>88</b>
<b>SUGGESTED PROJECTS</b> .....	<b>88</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>88</b>
<b>REFERENCES</b> .....	<b>89</b>
<b>NOTES</b> .....	<b>89</b>

## 4 COMPUTERS, PROGRAMS, AND ALGORITHMS

<b>INTRODUCTION</b> .....	<b>92</b>
<b>WHAT IS A COMPUTER?</b> .....	<b>92</b>
<i>COMPUTER VS. COMPUTER SYSTEM</i> .....	92
<i>COMPUTER SYSTEM</i> .....	92
<i>PROCESSOR</i> .....	94
<i>THREE ASPECTS OF PROCESSOR ARCHITECTURE</i> .....	95
<i>FEATURE SET</i> .....	95
<i>FEATURE SET IMPLEMENTATION</i> .....	95
<i>FEATURE SET ACCESSIBILITY</i> .....	95
<b>MEMORY ORGANIZATION</b> .....	<b>95</b>
<i>MEMORY BASICS</i> .....	96
<i>MEMORY HIERARCHY</i> .....	96
<i>Bits, BYTES, WORDS</i> .....	96
<i>ALIGNMENT AND ADDRESSABILITY</i> .....	97
<b>WHAT IS A PROGRAM?</b> .....	<b>98</b>
<i>TWO VIEWS OF A PROGRAM</i> .....	98
<i>THE HUMAN PERSPECTIVE</i> .....	98
<i>THE COMPUTER PERSPECTIVE</i> .....	98
<b>THE PROCESSING CYCLE</b> .....	<b>98</b>
<i>FETCH</i> .....	99
<i>DECODE</i> .....	99
<i>EXECUTE</i> .....	99
<i>STORE</i> .....	99
<i>Why A PROGRAM CRASHES</i> .....	99
<b>ALGORITHMS</b> .....	<b>99</b>
<i>Good vs. Bad Algorithms</i> .....	100
<i>DON'T REINVENT THE WHEEL!</i> .....	102
<b>THE JAVA HOTSPOT™ VIRTUAL MACHINE</b> .....	<b>102</b>
<i>OBTAINING THE JAVA HOTSPOT™ VIRTUAL MACHINE</i> .....	102
<i>CLIENT &amp; SERVER VIRTUAL MACHINES</i> .....	102
<i>CLASSIC VM vs. JIT vs. HOTSPOT™</i> .....	103
<i>JAVA HOTSPOT™ VIRTUAL MACHINE ARCHITECTURE</i> .....	103
<b>SERVICES PROVIDED BY THE JAVA HOTSPOT™ VIRTUAL MACHINE AND THE JRE</b> .....	<b>103</b>
<b>SUMMARY</b> .....	<b>104</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>104</b>
<b>SUGGESTED PROJECTS</b> .....	<b>104</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>105</b>
<b>REFERENCES</b> .....	<b>105</b>
<b>NOTES</b> .....	<b>105</b>

# PART II: LANGUAGE FUNDAMENTALS

## 5 OVERVIEW OF THE JAVA PLATFORM API

<b>INTRODUCTION</b> .....	<b>110</b>
<b>JAVA PLATFORM API PACKAGES</b> .....	<b>110</b>
<i>Obtaining Detailed Java Platform API Information</i> .....	<i>111</i>
<b>NAVIGATING A CLASS INHERITANCE HIERARCHY</b> .....	<b>112</b>
<i>Deprecated Methods</i> .....	<i>115</i>
<b>JAVA PLATFORM PACKAGES USED IN THIS BOOK</b> .....	<b>116</b>
<b>SUMMARY</b> .....	<b>117</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>117</b>
<b>SUGGESTED PROJECTS</b> .....	<b>118</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>118</b>
<b>REFERENCES</b> .....	<b>118</b>
<b>NOTES</b> .....	<b>119</b>

## 6 Simple Java Programs: Using Primitive And Reference Data Types

<b>INTRODUCTION</b> .....	<b>122</b>
<b>TERMINOLOGY</b> .....	<b>122</b>
<b>DEFINITION OF A JAVA PROGRAM</b> .....	<b>123</b>
<i>Application Objects</i> .....	<i>123</i>
<i>Talking About Applications</i> .....	<i>123</i>
<i>Applet Objects</i> .....	<i>123</i>
<b>CREATING SIMPLE JAVA PROGRAMS (APPLICATIONS)</b> .....	<b>123</b>
<i>STRUCTURE OF A JAVA APPLICATION</i> .....	<i>124</i>
<i>Compiling and Executing SimpleApplication</i> .....	<i>125</i>
<i>Quick Review</i> .....	<i>125</i>
<i>Building Bigger Applications</i> .....	<i>126</i>
<b>IDENTIFIERS AND IDENTIFIER NAMING RULES</b> .....	<b>126</b>
<i>Well-Named Identifiers Will Simplify Your Life</i> .....	<i>127</i>
<b>JAVA RESERVED KEYWORDS</b> .....	<b>127</b>
<b>JAVA TYPE CATEGORIES</b> .....	<b>129</b>
<i>PRIMITIVE DATA TYPES</i> .....	<i>129</i>
<i>REFERENCE DATA TYPES</i> .....	<i>130</i>
<i>Array Types</i> .....	<i>130</i>
<b>WORKING WITH PRIMITIVE TYPES</b> .....	<b>130</b>
<i>Application Class Structure</i> .....	<i>130</i>
<i>Definition Of The Term "Variable"</i> .....	<i>131</i>
<i>Declaring And Using Variables In The main() Method</i> .....	<i>131</i>
<i>Definition Of The Term "Constant"</i> .....	<i>132</i>
<i>Declaring And Using Constants In The main() Method</i> .....	<i>132</i>
<i>Getting Simple Input Into Your Program</i> .....	<i>133</i>
<i>Using The main() Method's String Array To Accept Input At Program Startup</i> .....	<i>133</i>
<b>WORKING WITH REFERENCE TYPES</b> .....	<b>134</b>
<i>Definition Of A Reference Variable</i> .....	<i>135</i>
<i>Definition Of An Object</i> .....	<i>135</i>
<i>Creating Objects with the new Operator</i> .....	<i>135</i>
<i>Garbage Collection</i> .....	<i>136</i>
<b>ACCESSING CLASS MEMBERS FROM THE main() METHOD</b> .....	<b>136</b>
<i>The main() Method Is A Static Method</i> .....	<i>137</i>
<i>Definition Of The Term Class Variable/Constant</i> .....	<i>137</i>

<i>Definition Of The Term Instance Variable/Constant</i> .....	137
<i>An Example</i> .....	137
<b>STATEMENTS And Expressions</b> .....	<b>138</b>
<b>OPERATORS</b> .....	<b>139</b>
<i>How To Read The Operator Table</i> .....	139
<i>Use Parentheses To Explicitly Force Precedence</i> .....	141
<i>Java Operators In Detail</i> .....	141
<i>Arithmetic Operators +, -, *, /</i> .....	141
<i>String Concatenation Operator +</i> .....	142
<i>Modulus (Integer Remainder) Operator %</i> .....	143
<i>Greater-Than/Less-Than Operators &gt;, &gt;=, &lt;, &lt;=</i> .....	143
<i>Equality And Inequality Operators ==, !=</i> .....	144
<i>Conditional AND and OR Operators &amp;&amp;,   </i> .....	144
<i>Boolean AND, OR, XOR, and NOT Operators &amp;,  , ^, !</i> .....	145
<i>Ternary Conditional Operator ?:</i> .....	145
<i>Left Shift and Right Shift Operators &lt;&lt;, &gt;&gt;, &gt;&gt;&gt;</i> .....	146
<i>instanceof Operator</i> .....	147
<i>Unary Prefix And Postfix Increment And Decrement Operators ++, -</i> .....	147
<i>Member Access Operator</i> .....	147
<i>Bitwise AND, OR, XOR, and Complement Operators &amp;,  , ^, ~</i> .....	148
<i>Combination Assignment Operators +=, -=, *=, /=, &lt;&lt;=, &gt;&gt;=, &gt;&gt;&gt;=, &amp;=,  =, ^=</i> .....	148
<b>JAVA Primitive Type Wrapper Classes</b> .....	<b>149</b>
<b>SUMMARY</b> .....	<b>150</b>
<b>Skill-Building Exercises</b> .....	<b>150</b>
<b>SUGGESTED PROJECTS</b> .....	<b>151</b>
<b>Self-Test Questions</b> .....	<b>152</b>
<b>REFERENCES</b> .....	<b>152</b>
<b>NOTES</b> .....	<b>153</b>

## 7 CONTROLLING THE FLOW OF PROGRAM EXECUTION

<b>INTRODUCTION</b> .....	<b>156</b>
<b>SELECTION STATEMENTS</b> .....	<b>156</b>
<i>if STATEMENT</i> .....	156
<i>Executing Code Blocks in if STATEMENTS</i> .....	157
<i>Executing Consecutive if STATEMENTS</i> .....	157
<i>if/else STATEMENT</i> .....	158
<i>Chained if/else STATEMENTS</i> .....	159
<i>switch STATEMENT</i> .....	159
<i>Nested Switch STATEMENT</i> .....	161
<i>Quick Review</i> .....	162
<b>ITERATION STATEMENTS</b> .....	<b>162</b>
<i>while STATEMENT</i> .....	162
<i>Personality of the while STATEMENT</i> .....	163
<i>do/while STATEMENT</i> .....	164
<i>Personality of the do/while STATEMENT</i> .....	164
<i>FOR STATEMENT</i> .....	165
<i>How The for STATEMENT is Related to the While STATEMENT</i> .....	165
<i>Personality of the for STATEMENT</i> .....	165
<i>Nesting Iteration STATEMENTS</i> .....	166
<i>Mixing Selection &amp; Iteration STATEMENTS: A Powerful Combination</i> .....	166
<i>Quick Review</i> .....	168
<b>BREAK AND CONTINUE</b> .....	<b>168</b>
<i>Unlabeled break</i> .....	168
<i>Labeled break</i> .....	169
<i>Unlabeled CONTINUE</i> .....	170

<i>Labeled CONTINUE</i> .....	170
<i>Quick Review</i> .....	171
<b>SELECTION AND ITERATION STATEMENT SELECTION TABLE</b> .....	<b>171</b>
<b>SUMMARY</b> .....	<b>173</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>173</b>
<b>SUGGESTED PROJECTS</b> .....	<b>175</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>176</b>
<b>REFERENCES</b> .....	<b>177</b>
<b>NOTES</b> .....	<b>177</b>

## 8 ARRAYS

<b>INTRODUCTION</b> .....	<b>180</b>
<b>WHAT IS AN ARRAY?</b> .....	<b>180</b>
<i>Specifying Array Types</i> .....	181
<i>Quick Review</i> .....	182
<b>FUNCTIONALITY PROVIDED BY JAVA ARRAY TYPES</b> .....	<b>182</b>
<i>Array-Type Inheritance Hierarchy</i> .....	182
<i>The java.lang.reflect.Array Class</i> .....	182
<i>Special Properties Of Java Arrays</i> .....	183
<i>Quick Review</i> .....	183
<b>CREATING AND USING SINGLE-DIMENSIONAL ARRAYS</b> .....	<b>183</b>
<i>Arrays Of Primitive Types</i> .....	183
<i>How Primitive Type Array Objects Are Arranged In Memory</i> .....	184
<i>Calling Object And Class Methods On Array References</i> .....	185
<i>Creating Single-Dimensional Arrays Using Array Literal Values</i> .....	186
<i>Differences Between Arrays Of Primitive Types And Arrays Of Reference Types</i> .....	187
<i>Single-dimensional Arrays In Action</i> .....	187
<i>Message Array</i> .....	188
<i>Calculating Averages</i> .....	191
<i>Histogram: Letter Frequency Counter</i> .....	192
<i>Quick Review</i> .....	193
<b>CREATING AND USING MULTIDIMENSIONAL ARRAYS</b> .....	<b>194</b>
<i>Multidimensional Array Declaration Syntax</i> .....	194
<i>Memory Representation Of A Two Dimensional Array</i> .....	196
<i>Creating Multidimensional Arrays Using Array Literals</i> .....	196
<i>Ragged Arrays</i> .....	197
<i>Multidimensional Arrays In Action</i> .....	198
<i>Weighted Grade Tool</i> .....	198
<i>Quick Review</i> .....	199
<b>THE MAIN() METHOD'S STRING ARRAY</b> .....	<b>200</b>
<i>Purpose And Use Of The main() Method's String Array</i> .....	200
<b>MANIPULATING ARRAYS WITH THE java.util.ARRAYS CLASS</b> .....	<b>201</b>
<b>JAVA API CLASSES USED IN THIS CHAPTER</b> .....	<b>202</b>
<b>SUMMARY</b> .....	<b>203</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>203</b>
<b>SUGGESTED PROJECTS</b> .....	<b>204</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>206</b>
<b>REFERENCES</b> .....	<b>207</b>
<b>NOTES</b> .....	<b>207</b>

## 9 TOWARD PROBLEM ABSTRACTION: CREATING NEW DATA TYPES

<b>INTRODUCTION</b> .....	<b>210</b>
<b>ABSTRACTION: Amplify THE ESSENTIAL – ELIMINATE THE IRRELEVANT</b> .....	<b>210</b>



<i>Abstraction Is The Art Of Programming</i> .....	210
<i>Where Problem Abstraction Fits Into The Development Cycle</i> .....	210
<i>Creating Your Own Data Types</i> .....	211
<i>Case-Study Project: Write A People Manager Program</i> .....	211
<i>Quick Review</i> .....	213
<b>The UML Class Diagram</b> .....	<b>213</b>
<i>Quick Review</i> .....	214
<b>Overview Of The Java Class Construct</b> .....	<b>214</b>
<i>Four Categories Of Class Members</i> .....	214
<i>Static Or Class-Wide Fields</i> .....	214
<i>Non-Static Or Instance Fields</i> .....	215
<i>Static Or Class-Wide Methods</i> .....	215
<i>Non-Static Or Instance Methods</i> .....	215
<i>Access Modifiers</i> .....	215
<i>public</i> .....	215
<i>private</i> .....	215
<i>protected</i> .....	216
<i>package</i> .....	216
<i>The Concepts Of Horizontal Access, Interface, And Encapsulation</i> .....	216
<i>Quick Review</i> .....	217
<b>Methods</b> .....	<b>217</b>
<i>Method Naming – Use Action Words That Indicate The Method’s Purpose</i> .....	217
<i>Maximize Method Cohesion</i> .....	217
<i>Structure Of A Method Definition</i> .....	217
<i>Method Modifiers (optional)</i> .....	218
<i>Return Type Or void (optional)</i> .....	219
<i>Method Name (Mandatory)</i> .....	219
<i>Parameter List (optional)</i> .....	219
<i>Method Body (optional for abstract or native methods)</i> .....	219
<i>Example Method Definitions</i> .....	219
<i>Method Signatures</i> .....	220
<i>Overloading Methods</i> .....	220
<i>Constructor Methods</i> .....	220
<i>Quick Review</i> .....	220
<b>Building And Testing The Person Class</b> .....	<b>221</b>
<i>Start By Creating The Source File And Class Definition Shell</i> .....	221
<i>Defining Person Instance Fields</i> .....	221
<i>Defining Person Instance Methods</i> .....	221
<i>The Constructor Method</i> .....	222
<i>Adding A Few Accessor Methods</i> .....	222
<i>Testing The Person Class</i> .....	223
<i>Use The PeopleManagerApplication Class As A Test Driver</i> .....	223
<i>Adding Features To The Person Class – Calculating Age</i> .....	224
<i>Adding Features To The Person Class – Convenience Methods</i> .....	225
<i>Adding Features To The Person Class – Finishing Touches</i> .....	226
<i>Quick Review</i> .....	227
<b>Building And Testing The PeopleManager Class</b> .....	<b>228</b>
<i>Defining The PeopleManager Class Shell</i> .....	228
<i>Defining PeopleManager Fields</i> .....	228
<i>Defining PeopleManager Constructor Methods</i> .....	228
<i>Defining Additional PeopleManager Methods</i> .....	229
<i>Testing The PeopleManager Class</i> .....	230
<i>Adding Features To The PeopleManager Class</i> .....	230
<i>Quick Review</i> .....	232
<b>More About Methods</b> .....	<b>232</b>
<i>How Arguments Are Passed To Methods</i> .....	232
<i>Local Variable Scoping</i> .....	233
<i>Anywhere An Object Of &lt;type&gt; Is Required, A Method That Returns &lt;type&gt; Can Be Used</i> .....	233

<i>Quick Review</i> .....	234
<b>Special Topics</b> .....	<b>234</b>
<i>Static Initializers</i> .....	234
<i>Data Encapsulation - The Naked Truth</i> .....	235
<i>Quick Review</i> .....	236
<b>SUMMARY</b> .....	<b>236</b>
<b>Skill-Building Exercises</b> .....	<b>237</b>
<b>SUGGESTED PROJECTS</b> .....	<b>238</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>239</b>
<b>REFERENCES</b> .....	<b>241</b>
<b>NOTES</b> .....	<b>241</b>

## 10 Compositional Design

<b>INTRODUCTION</b> .....	<b>244</b>
<b>MANAGING CONCEPTUAL AND PHYSICAL COMPLEXITY</b> .....	<b>244</b>
<i>Compiling Multiple Source Files Simultaneously with javac</i> .....	244
<i>Quick Review</i> .....	245
<b>DEPENDENCY VS. ASSOCIATION</b> .....	<b>245</b>
<b>AGGREGATION</b> .....	<b>245</b>
<i>Simple vs. Composite Aggregation</i> .....	246
<i>The Relationship Between Aggregation and Object Lifetime</i> .....	246
<i>Quick Review</i> .....	246
<b>EXPRESSING AGGREGATION IN A UML CLASS DIAGRAM</b> .....	<b>246</b>
<i>Simple Aggregation</i> .....	246
<i>Composite Aggregation</i> .....	247
<b>AGGREGATION EXAMPLE CODE</b> .....	<b>247</b>
<i>Simple Aggregation Example</i> .....	248
<i>Composite Aggregation Example</i> .....	249
<i>Quick Review</i> .....	250
<b>SEQUENCE DIAGRAMS</b> .....	<b>250</b>
<i>Quick Review</i> .....	251
<b>THE AIRCRAFT ENGINE SIMULATION: AN EXTENDED EXAMPLE</b> .....	<b>251</b>
<i>The Purpose of the Engine Class</i> .....	252
<i>Engine Class Attributes and Methods</i> .....	252
<i>PartStatus - A Typesafe Enumeration Pattern</i> .....	254
<i>Aircraft Engine Simulation Sequence Diagrams</i> .....	254
<b>RUNNING THE AIRCRAFT ENGINE SIMULATION PROGRAM</b> .....	<b>256</b>
<i>Quick Review</i> .....	256
<b>COMPLETE AIRCRAFT ENGINE SIMULATION CODE LISTING</b> .....	<b>257</b>
<b>SUMMARY</b> .....	<b>260</b>
<b>Skill-Building Exercises</b> .....	<b>260</b>
<b>SUGGESTED PROJECTS</b> .....	<b>261</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>262</b>
<b>REFERENCES</b> .....	<b>263</b>
<b>NOTES</b> .....	<b>263</b>

## 11 Extending Class Behavior Through Inheritance

<b>INTRODUCTION</b> .....	<b>266</b>
<b>THREE PURPOSES OF INHERITANCE</b> .....	<b>266</b>
<i>Implementing The "is A" Relationship</i> .....	267
<i>The Relationship Between The Terms Type, Interface, and Class</i> .....	267
<i>Meaning Of The Term Interface</i> .....	267

<i>MEANING OF THE TERM CLASS</i> .....	267
<i>Quick Review</i> .....	268
<b>EXPRESSING GENERALIZATION &amp; SPECIALIZATION IN THE UML</b> .....	<b>268</b>
<b>A SIMPLE INHERITANCE EXAMPLE</b> .....	<b>269</b>
<i>THE UML DIAGRAM</i> .....	269
<i>BASECLASS SOURCE CODE</i> .....	269
<i>DERIVEDCLASS SOURCE CODE</i> .....	270
<i>DRIVERAPPLICATION PROGRAM</i> .....	270
<i>Quick Review</i> .....	271
<b>ANOTHER INHERITANCE EXAMPLE: PERSON - STUDENT</b> .....	<b>271</b>
<i>THE PERSON - STUDENT UML CLASS DIAGRAM</i> .....	271
<i>PERSON - STUDENT SOURCE CODE</i> .....	272
<i>CASTING</i> .....	274
<i>USE CASTING SPARINGLY</i> .....	275
<i>Quick Review</i> .....	275
<b>OVERRIDING BASE CLASS METHODS</b> .....	<b>275</b>
<i>Quick Review</i> .....	276
<b>ABSTRACT METHODS &amp; ABSTRACT BASE CLASSES</b> .....	<b>276</b>
<i>THE PRIMARY PURPOSE OF AN ABSTRACT BASE CLASS</i> .....	276
<i>EXPRESSING ABSTRACT BASE CLASSES IN UML</i> .....	277
<i>Quick Review</i> .....	279
<b>INTERFACES</b> .....	<b>279</b>
<i>THE PURPOSE OF INTERFACES</i> .....	279
<i>AUTHORIZED INTERFACE MEMBERS</i> .....	279
<i>THE DIFFERENCES BETWEEN AN INTERFACE AND AN ABSTRACT CLASS</i> .....	280
<i>EXPRESSING INTERFACES IN UML</i> .....	280
<i>EXPRESSING REALIZATION IN A UML CLASS DIAGRAM</i> .....	280
<i>AN INTERFACE EXAMPLE</i> .....	281
<i>Quick Review</i> .....	282
<b>CONTROLLING HORIZONTAL &amp; VERTICAL ACCESS</b> .....	<b>283</b>
<i>Quick Review</i> .....	285
<b>FINAL CLASSES &amp; METHODS</b> .....	<b>285</b>
<i>Quick Review</i> .....	285
<b>POLYMORPHIC BEHAVIOR</b> .....	<b>285</b>
<i>Quick Review</i> .....	285
<b>INHERITANCE EXAMPLE: EMPLOYEE</b> .....	<b>286</b>
<b>INHERITANCE EXAMPLE: AIRCRAFT ENGINE SIMULATION</b> .....	<b>288</b>
<i>AIRCRAFT ENGINE SIMULATION UML DIAGRAM</i> .....	288
<i>SIMULATION OPERATIONAL DESCRIPTION</i> .....	290
<i>TAKE A DEEP BREATH AND RELAX!</i> .....	290
<i>COMPILING THE AIRCRAFT ENGINE SIMULATION CODE</i> .....	290
<b>COMPLETE AIRCRAFT ENGINE SIMULATION CODE LISTING</b> .....	<b>290</b>
<b>TERMS &amp; DEFINITIONS</b> .....	<b>297</b>
<b>SUMMARY</b> .....	<b>297</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>298</b>
<b>SUGGESTED PROJECTS</b> .....	<b>299</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>300</b>
<b>REFERENCES</b> .....	<b>301</b>
<b>NOTES</b> .....	<b>302</b>

# PART III: GRAPHICAL USER INTERFACE PROGRAMMING

## 12 JAVA SWING API OVERVIEW

<b>INTRODUCTION</b> .....	<b>306</b>
<b>AWT and SWING</b> .....	<b>306</b>
<i>NAMING CONVENTIONS</i> .....	307
<b>THE MATHEMATICS of GUIs</b> .....	<b>307</b>
<i>COORDINATE SYSTEMS</i> .....	307
<i>COMPONENTS AND THEIR BOUNDS</i> .....	308
<b>TOP-LEVEL CONTAINERS</b> .....	<b>308</b>
<i>WINDOW AND JWINDOW</i> .....	309
<i>FRAME AND JFRAME</i> .....	309
<i>DIALOG AND JDialog</i> .....	309
<i>OUR FIRST GUI PROGRAM</i> .....	310
<i>EXPLANATION OF THE CODE</i> .....	311
<i>TOP-LEVEL CONTAINERS API</i> .....	312
<i>TOP-LEVEL CONTAINER CONSTRUCTORS</i> .....	312
<i>DIALOGUE WITH A SKEPTIC</i> .....	313
<i>TOP-LEVEL CONTAINER METHODS</i> .....	314
<i>QUICK REVIEW</i> .....	316
<b>ORGANIZING COMPONENTS INTO CONTAINERS</b> .....	<b>316</b>
<i>ABSOLUTE POSITIONING</i> .....	316
<i>LAYOUT MANAGERS</i> .....	318
<i>FlowLayout</i> .....	318
<i>GridLayout</i> .....	320
<i>BorderLayout</i> .....	321
<i>GridBagLayout</i> .....	323
<i>COMBINING LAYOUTMANAGERS USING JPanels</i> .....	327
<i>QUICK REVIEW</i> .....	328
<b>THE COMPONENTS</b> .....	<b>329</b>
<i>BRIEF DESCRIPTIONS</i> .....	329
<i>COMPONENT, CONTAINER, AND JComponent METHODS</i> .....	331
<i>Component Method Tables</i> .....	332
<i>Container Method Tables</i> .....	335
<i>JComponent Method Tables</i> .....	336
<i>QUICK REVIEW</i> .....	337
<b>THE FINAL GUI</b> .....	<b>337</b>
<i>HIGHLIGHTS OF THE FINAL GUI</i> .....	341
<i>LAYOUT OF THE FINAL GUI</i> .....	342
<b>SUMMARY</b> .....	<b>344</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>345</b>
<b>SUGGESTED PROJECTS</b> .....	<b>346</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>346</b>
<b>REFERENCES</b> .....	<b>347</b>
<b>NOTES</b> .....	<b>347</b>

## 13 HANDLING GUI EVENTS

<b>INTRODUCTION</b> .....	<b>350</b>
<b>EVENT-HANDLING BASICS</b> .....	<b>350</b>
<i>THE EVENT</i> .....	350
<i>THE EVENT LISTENER</i> .....	351
<i>REGISTERING AN EVENT LISTENER WITH A COMPONENT</i> .....	351

Using the API .....	352
Choosing the Right Event .....	352
Quick Review .....	354
<b>CREATING EVENT LISTENERS .....</b>	<b>354</b>
LISTENINGMainFrame0 .....	354
LISTENINGMainFrame1 .....	355
STEP 1: APPLICATION BEHAVIOR .....	355
STEP 2: THE SOURCE .....	355
STEP 3: THE EVENT .....	355
STEP 4: THE LISTENER .....	357
STEP 5: THE DESIGN APPROACH: EXTERNAL CLASS .....	358
STEP 6: REGISTRATION .....	358
LISTENINGMainFrame2 .....	358
STEP 1: APPLICATION BEHAVIOR .....	359
STEP 2: THE SOURCE .....	359
STEP 3: THE EVENT .....	359
STEP 4: THE LISTENER .....	361
STEP 5: THE DESIGN APPROACH: EXTERNAL CLASS WITH FIELDS .....	362
STEP 6: REGISTRATION .....	363
LISTENINGMainFrame3 .....	364
STEP 1: APPLICATION BEHAVIOR .....	364
STEP 2: THE SOURCE .....	364
STEP 3: THE EVENT .....	364
STEP 4: THE LISTENER .....	365
STEP 5: THE DESIGN APPROACH: INNER CLASS (FIELD-LEVEL) .....	366
STEP 6: REGISTRATION .....	367
LISTENINGMainFrame4 .....	367
STEP 1: APPLICATION BEHAVIOR .....	367
STEP 2: THE SOURCE .....	367
STEP 3: THE EVENT .....	367
STEP 4: THE LISTENER .....	367
STEP 5: THE DESIGN APPROACH: INNER CLASS (LOCAL-VARIABLE-LEVEL) .....	368
STEP 6: REGISTRATION .....	368
LISTENINGMainFrame5 .....	368
STEP 1: APPLICATION BEHAVIOR .....	369
STEP 2: THE SOURCE .....	369
STEP 3: THE EVENT .....	369
STEP 4: THE LISTENER .....	369
STEP 5: THE DESIGN APPROACH: EXISTING CLASS .....	369
STEP 6: REGISTRATION .....	371
LISTENINGMainFrame6 .....	371
STEP 1: APPLICATION BEHAVIOR .....	371
STEP 2: THE SOURCE .....	371
STEP 3: THE EVENT .....	371
STEP 4: THE LISTENER .....	372
STEP 5: THE DESIGN APPROACH: ANONYMOUS CLASS .....	372
STEP 6: REGISTRATION .....	372
LISTENINGMainFrame7 .....	373
STEP 1: APPLICATION BEHAVIOR .....	373
STEP 2: THE SOURCE .....	373
STEP 3: THE EVENT .....	373
STEP 4: THE LISTENER .....	373
STEP 5: THE DESIGN APPROACH: ANONYMOUS CLASS AND INSTANCE .....	374
STEP 6: REGISTRATION .....	374
<b>ADAPTERS .....</b>	<b>374</b>
<b>SUMMARY .....</b>	<b>375</b>
<b>SKILL-BUILDING EXERCISES .....</b>	<b>376</b>
<b>SUGGESTED PROJECTS .....</b>	<b>376</b>

<b>SELF-TEST QUESTIONS</b> .....	<b>377</b>
<b>REFERENCES</b> .....	<b>377</b>
<b>NOTES</b> .....	<b>377</b>

## 14 AN ADVANCED GUI PROJECT

<b>INTRODUCTION</b> .....	<b>380</b>
<i>THE PROJECT</i> .....	380
<i>THE APPROACH</i> .....	380
<b>STEP 1 – LAYING THE GROUNDWORK</b> .....	<b>381</b>
<i>THE GARMENT CLASS</i> .....	381
<i>THE DRESSINGBOARD CLASS</i> .....	382
<i>GRAPHICS AND THE AWT/SWING FRAMEWORK</i> .....	382
<i>GRAPHICS (THE “HOW” OF DRAWING)</i> .....	382
<i>THE AWT/SWING FRAMEWORK (THE “WHEN” OF DRAWING)</i> .....	384
<i>CALLING REPAINT</i> .....	385
<i>OBTAINING A GRAPHICS2D</i> .....	386
<i>THE DRESSINGBOARD CLASS (CONTINUED)</i> .....	386
<i>LOADING AN IMAGE OR RESOURCE</i> .....	387
<i>THE MAINFRAME CLASS</i> .....	388
<i>QUICK REVIEW</i> .....	389
<b>STEP 2 – SWING’S SEPARABLE MODEL ARCHITECTURE</b> .....	<b>389</b>
<i>USING A LISTMODEL</i> .....	389
<i>THE MAINFRAME CLASS</i> .....	390
<i>QUICK REVIEW</i> .....	392
<b>STEP 3 – WRITING A CUSTOM RENDERER</b> .....	<b>392</b>
<i>USING A RENDERER</i> .....	393
<i>THE CHECKBOXLISTCELL CLASS</i> .....	394
<i>THE MAINFRAME CLASS</i> .....	396
<i>QUICK REVIEW</i> .....	398
<b>STEP 4 – THE ART OF ILLUSION</b> .....	<b>398</b>
<i>THE CHECKBOXLISTCELL CLASS</i> .....	401
<i>THE MAINFRAME CLASS</i> .....	402
<i>QUICK REVIEW</i> .....	403
<b>STEP 5 – DEFINING YOUR OWN EVENTS AND EVENTLISTENERS</b> .....	<b>403</b>
<i>THE CHECKBOXLISTCELL CLASS</i> .....	403
<i>THE MAINFRAME CLASS</i> .....	406
<i>QUICK REVIEW</i> .....	408
<b>INTERLUDE – WRITING A CUSTOM EDITOR</b> .....	<b>408</b>
<i>THE DEMOTREEORTABLECELLHANDLER CLASS</i> .....	410
<i>THE DEMOFRAME CLASS</i> .....	412
<i>QUICK REVIEW</i> .....	413
<b>STEP 6 – A GOOD COMPONENT GETS BETTER</b> .....	<b>413</b>
<i>THE DRAGLIST CLASS – OVERVIEW</i> .....	414
<i>ALGORITHM</i> .....	414
<i>USE DEFAULTLISTMODEL (LINES 41 - 65)</i> .....	414
<i>REORDEREVENT AND REORDERLISTENER (LINES 184 - 203)</i> .....	414
<i>THE DRAGLIST CLASS – DRAGGING</i> .....	415
<i>DRAGINDEX, DRAGITEM AND DRAGRECT ATTRIBUTES</i> .....	415
<i>DRAGSTART, DRAGTHRESHOLD AND ALLOWDRAG ATTRIBUTES</i> .....	415
<i>DELTA Y AND INDRAG ATTRIBUTES</i> .....	415
<i>HOW THE INDRAG ATTRIBUTE IS USED</i> .....	416
<i>THE DRAGLIST CLASS – PAINTING</i> .....	416
<i>THE CREATEDRAGIMAGE() METHOD (LINES 66 - 94)</i> .....	416
<i>THE PAINTCOMPONENT() METHOD (LINES 95 - 108)</i> .....	417
<i>THE DRAGLIST CLASS – CODE</i> .....	417
<i>THE MAINFRAME CLASS</i> .....	419

<b>SUMMARY</b> .....	<b>421</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>422</b>
<b>SUGGESTED PROJECTS</b> .....	<b>422</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>423</b>
<b>REFERENCES</b> .....	<b>423</b>
<b>NOTES</b> .....	<b>423</b>

## PART IV: INTERMEDIATE CONCEPTS

### 15 EXCEPTIONS

<b>INTRODUCTION</b> .....	<b>428</b>
<b>THE THROWABLE CLASS HIERARCHY</b> .....	<b>428</b>
<i>Errors</i> .....	429
<i>Exceptions</i> .....	429
<i>RuntimeExceptions</i> .....	429
<i>Checked vs. Unchecked Exceptions</i> .....	430
<i>Quick Review</i> .....	430
<b>HANDLING EXCEPTIONS</b> .....	<b>430</b>
<i>Try/Catch Statement</i> .....	431
<i>Multiple Catch Blocks</i> .....	431
<i>The Importance Of Exception Type And Order In Multiple Catch Blocks</i> .....	432
<i>Manipulating A Throwable Object</i> .....	433
<i>Try/Catch/Finally Statement</i> .....	435
<i>Try/Finally Statement</i> .....	435
<i>Quick Review</i> .....	436
<b>THROWING EXCEPTIONS</b> .....	<b>436</b>
<i>The Throws Clause</i> .....	436
<i>The Throw Keyword</i> .....	437
<i>Quick Review</i> .....	438
<b>CREATING CUSTOM EXCEPTIONS</b> .....	<b>438</b>
<i>High &amp; Low Level Abstractions</i> .....	438
<i>Extending The Exception Class</i> .....	438
<i>Some Advice On Naming Your Custom Exceptions</i> .....	440
<i>Quick Review</i> .....	440
<b>SUMMARY</b> .....	<b>440</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>441</b>
<b>SUGGESTED PROJECTS</b> .....	<b>441</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>441</b>
<b>REFERENCES</b> .....	<b>441</b>
<b>NOTES</b> .....	<b>442</b>

### 16 THREADS

<b>INTRODUCTION</b> .....	<b>444</b>
<b>WHAT IS A THREAD?</b> .....	<b>444</b>
<i>Quick Review</i> .....	446
<b>BUILDING A CLOCK COMPONENT</b> .....	<b>446</b>
<i>currentThread(), sleep(), interrupt(), interrupted() and isInterrupted()</i> .....	447
<i>Creating Your Own Threads</i> .....	449
<i>Quick Review</i> .....	451

<b>Computing Pi .....</b>	<b>452</b>
<i>Quick Review .....</i>	455
<b>Thread Priority and Scheduling .....</b>	<b>455</b>
<i>Your Results May Vary .....</i>	457
<i>Quick Review .....</i>	458
<b>Race Conditions .....</b>	<b>459</b>
<i>The Mechanics of Synchronization .....</i>	460
<i>Three Basic Rules of Synchronization .....</i>	462
<i>Synchronization Rule 1.....</i>	462
<i>Synchronization Rule 2.....</i>	463
<i>Synchronization Rule 3.....</i>	465
<i>Synchronizing Methods .....</i>	466
<i>Quick Review .....</i>	467
<b>The Producer-Consumer Relationship .....</b>	<b>468</b>
<i>Quick Review .....</i>	473
<b>Deadlock .....</b>	<b>474</b>
<i>Quick Review .....</i>	476
<b>About the Pi-Generating Algorithm .....</b>	<b>477</b>
<b>Summary .....</b>	<b>477</b>
<b>Skill-Building Exercises .....</b>	<b>478</b>
<b>Suggested Projects .....</b>	<b>479</b>
<b>Self-Test Questions .....</b>	<b>479</b>
<b>References .....</b>	<b>480</b>
<b>Notes .....</b>	<b>480</b>

## 17 Collections

<b>Introduction .....</b>	<b>482</b>
<b>Case Study: Building A Dynamic Array .....</b>	<b>482</b>
<i>Evaluating DynamicArray .....</i>	484
<i>The ArrayList Class To The Rescue .....</i>	484
<i>The Power Of Collections – Polymorphic Code .....</i>	485
<b>Java Collections Framework Overview .....</b>	<b>486</b>
<i>The Purpose Of The Collections Framework .....</i>	486
<i>Framework Organization .....</i>	486
<i>Core Collections Interfaces.....</i>	486
<i>General Purpose Implementation Classes.....</i>	487
<i>Mapping An Implementation Class To Its Underlying Data Structure.....</i>	492
<i>Algorithms.....</i>	492
<i>Infrastructure.....</i>	493
<i>Quick Review .....</i>	493
<b>Java 1.4.2 Style Collections .....</b>	<b>493</b>
<i>Creating A Set From A List .....</i>	494
<i>PeopleManager Revisited .....</i>	494
<i>Casting Retrieved Objects .....</i>	496
<i>Creating New Data Structures From Existing Collections .....</i>	497
<i>Quick Review .....</i>	499
<b>Java 5 Style Collections: Generics .....</b>	<b>499</b>
<i>Java 5 Collection Framework Core Interfaces .....</i>	499
<i>Java 5 Collections Framework Sample Programs .....</i>	500
<i>SetTestApp Program Revised.....</i>	500
<i>When To Use The Enhanced For Loop.....</i>	501
<i>Static Polymorphism – Generic Methods .....</i>	501
<i>Quick Review .....</i>	502
<i>Moving Forward .....</i>	502
<b>Summary .....</b>	<b>503</b>



<b>Skill-Building Exercises</b> .....	<b>503</b>
<b>Suggested Projects</b> .....	<b>504</b>
<b>Self-Test Questions</b> .....	<b>504</b>
<b>References</b> .....	<b>505</b>
<b>Notes</b> .....	<b>505</b>

## 18 File I/O

<b>Introduction</b> .....	<b>508</b>
<b>JAVA I/O PACKAGE OVERVIEW</b> .....	<b>508</b>
<i>File</i> Class .....	508
<i>InputStream</i> Classes .....	508
<i>OutputStream</i> Classes .....	510
Reader Classes .....	510
Writer Classes .....	510
<i>RandomAccessFile</i> Class .....	510
How Do You Choose Between Byte Streams, Character Streams, and <i>RandomAccessFile</i> ? .....	510
Another Way To Categorize The Java I/O Classes .....	510
Quick Review .....	512
<b>Using The File Class</b> .....	<b>513</b>
Quick Review .....	515
<b>SEQUENTIAL BYTE STREAM FILE I/O USING INPUT- &amp; OUTPUTSTREAMS</b> .....	<b>515</b>
<i>OutputStreams</i> .....	515
<i>FileOutputStream</i> .....	515
<i>BufferedOutputStream</i> .....	516
<i>DataOutputStream</i> .....	517
<i>ObjectOutputStream</i> .....	518
<i>PrintStream</i> .....	520
<i>InputStreams</i> .....	520
<i>FileInputStream</i> .....	521
<i>BufferedInputStream</i> .....	521
<i>DataInputStream</i> .....	522
<i>ObjectInputStream</i> .....	523
Quick Review .....	523
<b>SEQUENTIAL CHARACTER STREAM FILE I/O USING READERS &amp; WRITERS</b> .....	<b>524</b>
Writers .....	524
<i>FileWriter</i> .....	524
<i>BufferedWriter</i> .....	525
<i>OutputStreamWriter</i> .....	526
<i>PrintWriter</i> .....	526
Readers .....	527
<i>FileReader</i> .....	527
<i>InputStreamReader</i> .....	527
<i>BufferedReader</i> .....	528
Quick Review .....	529
<b>A PRACTICAL FILE I/O EXAMPLE: THE <code>java.util.Properties</code> CLASS</b> .....	<b>529</b>
Quick Review .....	531
<b>RANDOM ACCESS FILE I/O</b> .....	<b>531</b>
Towards An Approach To The Adapter Project .....	533
Start Small And Take Baby Steps.....	533
Other Project Considerations .....	534
Locking A Record For Updates And Deletes .....	534
Translating Low-Level Exceptions Into Higher-Level Exception Abstractions.....	534
Where To Go From Here .....	534
Complete <i>RandomAccessFile</i> Legacy <i>Datafile</i> Adapter Source Code Listing .....	535
Quick Review .....	544

<b>Quick File I/O COMBINATION REFERENCE .....</b>	<b>545</b>
<b>SUMMARY .....</b>	<b>546</b>
<b>SKILL-BUILDING EXERCISES .....</b>	<b>547</b>
<b>SUGGESTED PROJECTS .....</b>	<b>547</b>
<b>SELF-TEST QUESTIONS .....</b>	<b>548</b>
<b>REFERENCES .....</b>	<b>549</b>
<b>NOTES .....</b>	<b>549</b>

## PART V: NETWORK PROGRAMMING

### 19 INTRODUCTION TO NETWORKING AND DISTRIBUTED APPLICATIONS

<b>INTRODUCTION .....</b>	<b>554</b>
<b>WHAT IS A COMPUTER NETWORK? .....</b>	<b>554</b>
<i>PURPOSE OF A NETWORK .....</i>	<i>554</i>
<i>THE ROLE OF NETWORK PROTOCOLS .....</i>	<i>555</i>
<i>HOMOGENEOUS VS. HETEROGENEOUS NETWORKS .....</i>	<i>555</i>
<i>THE UNIFYING NETWORK PROTOCOLS: TCP/IP .....</i>	<i>555</i>
<i>WHAT'S SO SPECIAL ABOUT THE INTERNET? .....</i>	<i>556</i>
<i>Quick Review .....</i>	<i>556</i>
<b>SERVERS &amp; CLIENTS .....</b>	<b>557</b>
<i>SERVER HARDWARE AND SOFTWARE .....</i>	<i>557</i>
<i>CLIENT HARDWARE AND SOFTWARE .....</i>	<i>557</i>
<i>Quick Review .....</i>	<i>558</i>
<b>Application Distribution .....</b>	<b>558</b>
<i>Physical Distribution ON ONE COMPUTER .....</i>	<i>558</i>
<i>RUNNING MULTIPLE JAVA VIRTUAL MACHINES ON THE SAME COMPUTER .....</i>	<i>558</i>
<i>STARTING MULTIPLE JVMs IN WINDOWS .....</i>	<i>558</i>
<i>STARTING MULTIPLE JVMs IN UNIX AND LINUX .....</i>	<i>559</i>
<i>RUNNING MULTIPLE CLIENTS ON THE SAME COMPUTER .....</i>	<i>561</i>
<i>ADDRESSING THE LOCAL MACHINE .....</i>	<i>561</i>
<i>Physical Distribution ACROSS MULTIPLE COMPUTERS .....</i>	<i>561</i>
<i>MULTIPLE JAVA VIRTUAL MACHINES .....</i>	<i>562</i>
<i>Quick Review .....</i>	<i>562</i>
<b>Multi-Tiered Applications .....</b>	<b>562</b>
<i>Logical Application TIERS .....</i>	<i>562</i>
<i>Physical Tier Distribution .....</i>	<i>563</i>
<i>Quick Review .....</i>	<i>564</i>
<b>INTERNET NETWORKING PROTOCOLS – NUTS &amp; BOLTS .....</b>	<b>564</b>
<i>THE INTERNET PROTOCOLS: TCP, UDP, AND IP .....</i>	<i>564</i>
<i>THE APPLICATION LAYER .....</i>	<i>564</i>
<i>TRANSPORT LAYER .....</i>	<i>565</i>
<i>NETWORK LAYER .....</i>	<i>565</i>
<i>DATA LINK &amp; PHYSICAL LAYERS .....</i>	<i>565</i>
<i>PUTTING IT ALL TOGETHER .....</i>	<i>566</i>
<i>WHAT YOU NEED TO KNOW .....</i>	<i>566</i>
<i>Quick Review .....</i>	<i>566</i>
<b>JAVA SUPPORT FOR NETWORK PROGRAMMING .....</b>	<b>567</b>
<i>A NETWORK PROGRAMMING SCENARIO USING SOCKETS – OVERVIEW .....</i>	<i>567</i>
<i>THE URL CLASS .....</i>	<i>568</i>
<i>Quick Review .....</i>	<i>569</i>
<b>REMOTE METHOD INVOCATION (RMI) OVERVIEW .....</b>	<b>569</b>
<i>CREATING &amp; RUNNING AN RMI APPLICATION IN JAVA 1.4.2 .....</i>	<i>570</i>

REQUIRED STEPS .....	570
STEP 1: DEFINE A REMOTE INTERFACE.....	570
STEP 2: DEFINE A REMOTE INTERFACE IMPLEMENTATION.....	571
STEP 3: COMPILE THE REMOTE INTERFACE & REMOTE INTERFACE IMPLEMENTATION SOURCE FILES.....	571
STEP 4: USE THE RMIC TOOL TO GENERATE STUB CLASS .....	571
STEP 5: CREATE SERVER APPLICATION.....	572
STEP 6: CREATE CLIENT APPLICATION.....	573
STEP 7: START THE RMI REGISTRY.....	573
STEP 8: RUN THE SERVER APPLICATION.....	574
STEP 9: RUN THE CLIENT APPLICATION.....	574
CREATING & RUNNING AN RMI APPLICATION IN JAVA 1.5 .....	575
CONSIDERATIONS WHEN DEPLOYING RMI APPLICATIONS IN MIXED JVM VERSION ENVIRONMENTS .....	575
SERVER JVM 1.5 - CLIENT JVM 1.5.....	575
SERVER JVM 1.5 - CLIENT JVM 1.4.2.....	575
SERVER JVM 1.4.2 - CLIENT JVM 1.5.....	575
SERVER JVM 1.4.2 - CLIENT JVM 1.4.2.....	575
Quick Review .....	576
<b>SUMMARY .....</b>	<b>576</b>
<b>SKILL-BUILDING EXERCISES .....</b>	<b>577</b>
<b>SUGGESTED PROJECTS .....</b>	<b>578</b>
<b>SELF-TEST QUESTIONS .....</b>	<b>578</b>
<b>REFERENCES .....</b>	<b>579</b>
<b>NOTES .....</b>	<b>579</b>

## 20 CLIENT-SERVER APPLICATIONS

<b>INTRODUCTION .....</b>	<b>582</b>
<b>A SIMPLE SOCKET-BASED CLIENT-SERVER EXAMPLE .....</b>	<b>582</b>
Simple SERVER Application .....	583
Simple CLIENT Application .....	585
RUNNING THE EXAMPLES .....	586
A FEW WORDS ABOUT PROTOCOL .....	588
Quick Review .....	588
<b>PROJECT SPECIFICATION .....</b>	<b>588</b>
<b>CLIENT-SERVER PROJECT INITIAL DESIGN CONSIDERATIONS .....</b>	<b>589</b>
NOUN-VERB ANALYSIS .....	589
HIGH-LEVEL APPLICATION OPERATIONS .....	590
HOW TO PROCEED .....	592
Quick Review .....	592
<b>IMPLEMENTING NETRATSERVER – FIRST ITERATION .....</b>	<b>592</b>
FIRST ITERATION CODE .....	593
TESTING THE FIRST ITERATION CODE .....	595
Quick Review .....	596
<b>IMPLEMENTING NETRATSERVER - SECOND ITERATION .....</b>	<b>596</b>
SECOND ITERATION CODE .....	597
TESTING SECOND ITERATION CODE .....	599
Quick Review .....	599
<b>THE RMI-BASED CLIENT .....</b>	<b>600</b>
TESTING MULTIPLE RMI CLIENTS .....	602
Quick Review .....	602
<b>NETRATSERVER IMPLEMENTATION – THIRD ITERATION .....</b>	<b>602</b>
UPDATED SERVER APPLICATION CLASS DIAGRAM .....	602
THIRD ITERATION CODE .....	603
TESTING THE THIRD ITERATION CODE .....	604
TESTING MULTIPLE RMI CLIENT APPLICATIONS .....	606
Quick Review .....	607

<b>NETRATSERVER IMPLEMENTATION – FOURTH ITERATION .....</b>	<b>607</b>
<i>FOURTH ITERATION CODE .....</i>	<i>608</i>
<i>TESTING FOURTH ITERATION CODE .....</i>	<i>610</i>
<i>TESTING THE FOURTH ITERATION CODE .....</i>	<i>613</i>
<i>QUICK REVIEW .....</i>	<i>613</i>
<b>NETRATSERVER IMPLEMENTATION – FINAL ITERATION .....</b>	<b>613</b>
<i>FINAL ITERATION CODE .....</i>	<i>615</i>
<i>TESTING THE FINAL ITERATION CODE .....</i>	<i>619</i>
<i>PARTING COMMENTS .....</i>	<i>619</i>
<i>QUICK REVIEW .....</i>	<i>620</i>
<b>SUMMARY .....</b>	<b>620</b>
<b>SKILL-BUILDING EXERCISES .....</b>	<b>621</b>
<b>SUGGESTED PROJECTS .....</b>	<b>622</b>
<b>SELF-TEST QUESTIONS .....</b>	<b>622</b>
<b>REFERENCES .....</b>	<b>623</b>
<b>NOTES .....</b>	<b>623</b>

## 21 Applets & JDBC

<b>INTRODUCTION .....</b>	<b>626</b>
<b>APPLET OVERVIEW .....</b>	<b>626</b>
<i>THE BENEFITS OF USING APPLETS .....</i>	<i>626</i>
<i>&lt;APPLET&gt; VS. &lt;OBJECT&gt; TAGS .....</i>	<i>627</i>
<i>BROWSER JVM VS. JAVA PLUG-IN .....</i>	<i>627</i>
<b>A BASIC APPLET EXAMPLE .....</b>	<b>627</b>
<i>APPLET CODE .....</i>	<i>627</i>
<i>HTML PAGE CODE .....</i>	<i>628</i>
<i>PACKAGING AND DISTRIBUTION .....</i>	<i>628</i>
<i>RUNNING THE BASIC APPLET EXAMPLE .....</i>	<i>628</i>
<i>APPLET LIFE-CYCLE STAGES.....</i>	<i>629</i>
<i>THE &lt;APPLET&gt; TAG IN DETAIL .....</i>	<i>630</i>
<i>QUICK REVIEW .....</i>	<i>631</i>
<b>APPLET SECURITY RESTRICTIONS .....</b>	<b>631</b>
<i>MISCELLANEOUS APPLET SECURITY RESTRICTIONS .....</i>	<i>634</i>
<i>SECURITY POLICIES &amp; SIGNED APPLETS .....</i>	<i>634</i>
<i>QUICK REVIEW .....</i>	<i>634</i>
<b>USING APPLET PARAMETERS .....</b>	<b>634</b>
<i>QUICK REVIEW .....</i>	<i>636</i>
<b>EXTENDED APPLET EXAMPLE – POETRY IN BROWSER .....</b>	<b>636</b>
<b>JAVA DATABASE CONNECTIVITY (JDBC™) OVERVIEW .....</b>	<b>639</b>
<i>JDBC™ PACKAGES .....</i>	<i>639</i>
<i>JDBC™ SPECIFICATION .....</i>	<i>639</i>
<i>USING JDBC – A MACRO VIEW .....</i>	<i>640</i>
<i>PROPERLY CONFIGURED DATABASE.....</i>	<i>640</i>
<i>JDBC DRIVER CLASS.....</i>	<i>640</i>
<i>KNOWLEDGE OF RELATIONAL DATABASE DESIGN .....</i>	<i>640</i>
<i>KNOWLEDGE OF THE DATABASE SCHEMA .....</i>	<i>640</i>
<i>KNOWLEDGE OF SQL.....</i>	<i>640</i>
<i>THE JAVA JDBC PROGRAM .....</i>	<i>640</i>
<i>GENERAL STEPS REQUIRED TO EMPLOY JDBC .....</i>	<i>641</i>
<i>MOVING FORWARD .....</i>	<i>641</i>
<i>QUICK REVIEW .....</i>	<i>641</i>
<b>JDBC PROJECT DESCRIPTION .....</b>	<b>641</b>
<i>OVERALL SYSTEM ARCHITECTURE .....</i>	<i>641</i>
<i>DETAILED SYSTEM ARCHITECTURE .....</i>	<i>641</i>
<i>ROLE OF THE PERSISTER CLASS .....</i>	<i>643</i>

Quick Review .....	643
<b>MySQL DATABASE &amp; JDBC .....</b>	<b>643</b>
Setting Up MySQL To Run The Example Code .....	643
Obtaining And Installing MySQL .....	644
The mysqladmin Program.....	644
The mysql Monitor Program.....	645
Helpful MySQL Monitor Commands.....	646
Creating The chapter_21 Database .....	647
Establishing Database Security Via MySQL Access Control Tables.....	647
General Strategy For Managing Permissions Using The MySQL Access Control Tables .....	649
Creating The Employee Training Management System Tables.....	650
Populating Tables with Data and Running Queries .....	651
Inserting Data Into Tables.....	651
Querying A Table with the select Statement.....	652
Updating Data In A Table.....	652
Joining Tables With A Select Statement.....	652
Accessing chapter_21 Database Tables Via JDBC .....	654
A Short JDBC Example Program.....	654
Accessing ResultSet Metadata.....	655
Quick Review .....	656
<b>Extended Applet &amp; JDBC Example .....</b>	<b>656</b>
The Code .....	656
Compiling And Packaging The Applet Code For Distribution .....	667
Running The Example .....	668
Start MySQL Database Application.....	668
Start The DBServerApp Program.....	668
Access The employeetraining.html Page.....	668
Using The Employee Training Applet.....	668
Code Discussion .....	670
Employee And EmployeeTraining Classes.....	670
AddNewEmployeeDialog And AddTrainingRecordDialog Classes.....	671
EmployeeTrainingApplet Class .....	671
DBServerApp Class .....	671
Persistor Class.....	671
DBServerProperties .....	671
Statements vs. Prepared Statements .....	672
<b>SUMMARY .....</b>	<b>672</b>
<b>Skill-Building Exercises .....</b>	<b>673</b>
<b>Suggested Projects .....</b>	<b>674</b>
<b>Self-Test Questions .....</b>	<b>675</b>
<b>REFERENCES .....</b>	<b>675</b>
<b>NOTES .....</b>	<b>676</b>

## PART VI: OBJECT-ORIENTED DESIGN

### 22 INHERITANCE, COMPOSITION, INTERFACES, POLYMORPHISM

<b>INTRODUCTION .....</b>	<b>680</b>
<b>INHERITANCE VS. COMPOSITION: THE GREAT DEBATE .....</b>	<b>680</b>
What's The End Game? .....	681
Flexible Application Architectures.....	681
Modularity And Reliability.....	681
Architectural Stability Via Managed Dependencies .....	681

<i>Knowing When To Accept A Design That's Good Enough</i> .....	682
<i>Quick Review</i> .....	682
<b>INHERITANCE-BASED DESIGN</b> .....	<b>682</b>
<i>THREE GOOD REASONS TO USE INHERITANCE</i> .....	682
<i>AS A MEANS TO REASON ABOUT CODE BEHAVIOR</i> .....	682
<i>TO GAIN A MEASURE OF CODE REUSE</i> .....	682
<i>TO FACILITATE INCREMENTAL DEVELOPMENT</i> .....	682
<i>FORMS OF INHERITANCE: MEYER'S INHERITANCE TAXONOMY</i> .....	683
<i>COAD'S INHERITANCE CRITERIA</i> .....	684
<i>PERSON - EMPLOYEE EXAMPLE REVISITED</i> .....	685
<i>Quick Review</i> .....	685
<b>THE ROLE OF INTERFACES</b> .....	<b>686</b>
<i>REDUCING OR LIMITING INTERMODULE DEPENDENCIES</i> .....	686
<i>MODELING DOMINANT, COLLATERAL, AND DYNAMIC ROLES</i> .....	686
<i>DOMINANT ROLES</i> .....	687
<i>COLLATERAL ROLES</i> .....	687
<i>DYNAMIC ROLES</i> .....	687
<i>Quick Review</i> .....	687
<b>Applied Polymorphism</b> .....	<b>687</b>
<i>Quick Review</i> .....	688
<b>Composition-Based Design As A Force Multiplier</b> .....	<b>688</b>
<i>TWO TYPES OF AGGREGATION</i> .....	688
<i>POLYMORPHIC CONTAINMENT</i> .....	688
<i>AN EXTENDED EXAMPLE</i> .....	689
<i>Quick Review</i> .....	693
<b>SUMMARY</b> .....	<b>693</b>
<b>Skill-Building Exercises</b> .....	<b>694</b>
<b>SUGGESTED PROJECTS</b> .....	<b>695</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>696</b>
<b>REFERENCES</b> .....	<b>696</b>
<b>NOTES</b> .....	<b>697</b>

## 23 Well-Behaved Objects

<b>INTRODUCTION</b> .....	<b>700</b>
<b>CONSIDER OBJECT USAGE FROM THE START</b> .....	<b>700</b>
<i>Applying THE OBJECT USAGE SCENARIO CHECKLIST</i> .....	701
<i>Quick Review</i> .....	702
<b>OVERRIDING java.lang.Object METHODS</b> .....	<b>703</b>
<i>OVERRIDING toString()</i> .....	703
<i>OVERRIDING equals() AND hashCode()</i> .....	703
<i>equals() METHOD</i> .....	703
<i>TESTING PERSON.toString() AND PERSON.equals() METHODS</i> .....	705
<i>hashCode() METHOD</i> .....	705
<i>OVERRIDING hashCode() IN THE PERSON CLASS</i> .....	707
<i>TESTING PERSON.hashCode()</i> .....	707
<i>OVERRIDING clone()</i> .....	708
<i>SHALLOW COPY VS. DEEP COPY</i> .....	708
<i>OVERRIDING PERSON.clone()</i> .....	709
<i>OVERRIDING finalize() – AND AN ALTERNATIVE APPROACH</i> .....	710
<i>Quick Review</i> .....	710
<b>IMPLEMENTING THE java.lang.Comparable INTERFACE</b> .....	<b>711</b>
<i>Quick Review</i> .....	712
<b>Using PERSON OBJECTS IN A COLLECTION</b> .....	<b>713</b>
<b>IMPLEMENTING java.util.Comparator INTERFACE</b> .....	<b>713</b>
<i>Quick Review</i> .....	715

<b>IMPLEMENTING THE SERIALIZABLE INTERFACE</b> .....	<b>716</b>
<b>PERSON CLASS – THE FINAL VERSION</b> .....	<b>716</b>
<b>SUMMARY</b> .....	<b>717</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>718</b>
<b>SUGGESTED PROJECTS</b> .....	<b>718</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>718</b>
<b>REFERENCES</b> .....	<b>719</b>
<b>NOTES</b> .....	<b>719</b>

## 24 THREE DESIGN PRINCIPLES

<b>INTRODUCTION</b> .....	<b>722</b>
<b>THE PREFERRED CHARACTERISTICS OF AN OBJECT-ORIENTED ARCHITECTURE</b> .....	<b>722</b>
<i>EASY TO UNDERSTAND – (HOW DOES THIS THING WORK?)</i> .....	722
<i>EASY TO REASON ABOUT – (WHAT ARE THE EFFECTS OF CHANGE?)</i> .....	722
<i>EASY TO EXTEND – (WHERE DO I ADD FUNCTIONALITY?)</i> .....	722
<b>THE LISKOV SUBSTITUTION PRINCIPLE &amp; DESIGN BY CONTRACT</b> .....	<b>723</b>
<i>REASONING ABOUT THE BEHAVIOR OF SUPERTYPES AND SUBTYPES</i> .....	723
<i>RELATIONSHIP BETWEEN THE LSP AND DbC</i> .....	723
<i>THE COMMON GOAL OF THE LSP AND DbC</i> .....	723
<i>JAVA SUPPORT FOR THE LSP AND DbC</i> .....	723
<i>DESIGNING WITH THE LSP/DbC IN MIND</i> .....	724
<i>CLASS DECLARATIONS VIEWED AS BEHAVIOR SPECIFICATIONS</i> .....	724
<i>Quick Review</i> .....	724
<b>PRECONDITIONS, POSTCONDITIONS, AND CLASS INVARIANTS</b> .....	<b>724</b>
<i>CLASS INVARIANT</i> .....	724
<i>PRECONDITION</i> .....	724
<i>POSTCONDITION</i> .....	725
<i>AN EXAMPLE</i> .....	725
<i>A NOTE ON USING ASSERTIONS TO ENFORCE PRE- AND POSTCONDITIONS</i> .....	726
<i>USING INCREMENTER AS A BASE CLASS</i> .....	726
<i>CHANGING THE PRECONDITIONS OF DERIVED CLASS METHODS</i> .....	728
<i>CHANGING THE POSTCONDITIONS OF DERIVED CLASS METHODS</i> .....	731
<i>SPECIAL CASES OF PRECONDITIONS AND POSTCONDITIONS</i> .....	732
<i>METHOD ARGUMENT TYPES</i> .....	732
<i>METHOD RETURN TYPES</i> .....	735
<i>THREE RULES OF THE SUBSTITUTION PRINCIPLE</i> .....	735
<i>SIGNATURE RULE</i> .....	735
<i>METHODS RULE</i> .....	735
<i>PROPERTIES RULE</i> .....	735
<i>Quick Review</i> .....	735
<b>THE OPEN-CLOSED PRINCIPLE</b> .....	<b>735</b>
<i>ACHIEVING THE OPEN-CLOSED PRINCIPLE</i> .....	736
<i>AN OCP EXAMPLE</i> .....	736
<i>Quick Review</i> .....	739
<b>THE DEPENDENCY INVERSION PRINCIPLE</b> .....	<b>740</b>
<i>CHARACTERISTICS OF BAD SOFTWARE ARCHITECTURE</i> .....	740
<i>CHARACTERISTICS OF GOOD SOFTWARE ARCHITECTURE</i> .....	741
<i>SELECTING THE RIGHT ABSTRACTIONS TAKES EXPERIENCE</i> .....	741
<i>Quick Review</i> .....	741
<b>TERMS AND DEFINITIONS</b> .....	<b>741</b>
<b>SUMMARY</b> .....	<b>742</b>
<b>SKILL-BUILDING EXERCISES</b> .....	<b>743</b>
<b>SUGGESTED PROJECTS</b> .....	<b>743</b>
<b>SELF-TEST QUESTIONS</b> .....	<b>743</b>
<b>REFERENCES</b> .....	<b>744</b>

**NOTES ..... 744**

## 25 Helpful Design Patterns

**INTRODUCTION ..... 746**

**SOFTWARE DESIGN PATTERNS AND HOW THEY CAME TO BE ..... 746**

*WHAT EXACTLY IS A SOFTWARE DESIGN PATTERN?* ..... 746

*ORIGINS* ..... 746

*PATTERN SPECIFICATION* ..... 747

*APPLYING SOFTWARE DESIGN PATTERNS* ..... 747

*QUICK REVIEW* ..... 747

**THE SINGLETON PATTERN ..... 748**

*QUICK REVIEW* ..... 749

**THE FACTORY PATTERN ..... 749**

*THE DYNAMIC FACTORY* ..... 750

*ADVANTAGES OF THE DYNAMIC FACTORY PATTERN* ..... 751

*QUICK REVIEW* ..... 751

**THE MODEL-VIEW-CONTROLLER PATTERN ..... 751**

*POTENTIAL ISSUES WITH ONE ACTIONLISTENER* ..... 753

*QUICK REVIEW* ..... 753

**THE COMMAND PATTERN ..... 753**

*QUICK REVIEW* ..... 756

**A COMPREHENSIVE PATTERN-BASED EXAMPLE ..... 756**

*Code Listing By Package Name* ..... 756

*com.pulpfreepress.commands* ..... 756

*com.pulpfreepress.controller* ..... 759

*com.pulpfreepress.exceptions* ..... 760

*com.pulpfreepress.interfaces* ..... 760

*com.pulpfreepress.model* ..... 761

*com.pulpfreepress.utils* ..... 767

*com.pulpfreepress.view* ..... 769

**SUMMARY ..... 776**

**SKILL-BUILDING EXERCISES ..... 777**

**SUGGESTED PROJECTS ..... 777**

**SELF-TEST QUESTIONS ..... 777**

**REFERENCES ..... 778**

**NOTES ..... 778**

## APPENDICES

### Appendix A: Helpful Checklists And Tables

**PROJECT-APPROACH STRATEGY CHECK-OFF LIST ..... 781**

**DEVELOPMENT CYCLE ..... 782**

### Appendix B: ASCII Table

**ASCII TABLE ..... 783**