Table of Contents

Preface 15

About the Authors 15
Who Should Buy This Book? 16
Where to Find Answers to Review Questions and Exercises 16
How to Report Errata 16
Conventions Used in This Book 17

Chapter 1 How a Computer Works 19

1.1 Introduction 19
1.2 What is Hardware? 19
1.3 What is Software? 20
1.4 How a Computer Executes (Runs) a Program 20
1.5 Compilers and Interpreters 21
1.6 What is Source Code? 22
1.7 Review Questions: True/False 22
1.8 Review Questions: Multiple Choice 22
1.9 Review Questions 24

Chapter 2 C# and Integrated Development Environments (IDEs) 25

2.1 What is C#? 25
2.2 How C# Works 25
2.3 Integrated Development Environments 26
2.4 Visual Studio 26
2.5 How to Set Up Visual Studio Community 27
2.6 Review Questions 30

Chapter 3 Basic Algorithmic Concepts 31

3.1 What is an Algorithm? 31
3.2 The Algorithm for Making a Cup of Tea 31
3.3 Okay about Algorithms. But what is a Computer Program Anyway? 32
3.4 The Three Parties! 32
3.5 The Three Main Stages Involved in Creating an Algorithm 32
3.6 What are “Reserved Words”? 33
3.7 Your First C# Program 33
3.8 What is the Difference between a Syntax Error, a Logic Error, and a Runtime Error? 34
3.9 What Does “Debugging” Mean? 35
3.10 Commenting Your Code 35
3.11 Review Questions: True/False 37
3.12 Review Questions: Multiple Choice 37

3.13 Review Questions 38

Chapter 4 Variables and Constants 39

4.1 What is a Variable? 39
4.2 What is a Constant? 41
4.3 How Many Types of Variables and Constants Exist? 41
4.4 Rules for Naming Variables and Constants in C# 41
4.5 What Does the Phrase "Declare a Variable" Mean? 42
4.6 How to Declare Variables in C# 42
4.7 How to Declare Constants in C# 44
4.8 Review Questions: True/False 44
4.9 Review Questions: Multiple Choice 45
4.10 Review Exercises 46
4.11 Review Questions 46

Chapter 5 Handling Input and Output 47

5.1 Which Statement Outputs Messages and Results to a User's Screen? 47
5.2 How to Alter the Default Behavior of a `Write` Method 49
5.3 Which Statement Lets the User Enter Data? Which Statement Prompts the User to Enter Data? 51
5.4 Review Questions: True/False 53
5.5 Review Questions: Multiple Choice 53
5.6 Review Questions 53

Chapter 6 Operators 55

6.1 The Value Assignment Operator 55
6.2 Arithmetic Operators 56
6.3 What is the Precedence of Arithmetic Operators? 57
6.4 Compound Assignment Operators 58
Exercise 6.4.1 — Which C# Statements are Syntactically Correct? 58
Exercise 6.4.2 — Finding Variable Types 59
6.5 Incrementing/Decrementing Operators 59
6.6 String Operators 61
Exercise 6.6.1 — Concatenating Names 61
6.7 Review Questions: True/False 62
6.8 Review Questions: Multiple Choice 63
6.9 Review Exercises 64
6.10 Review Questions 66

Chapter 7 Using Visual Studio 67

7.1 Creating a New C# Project 67
7.2 Writing and Executing a C# Program 68
7.3 Debugging C# Programs with Visual Studio 72

Chapter 8 Writing your First Real Programs 77
8.1 Introduction 77
Exercise 8.1.1 — Calculating the Area of a Rectangle 78
Exercise 8.1.2 — Calculating the Area of a Circle 78
Exercise 8.1.3 — Fahrenheit to Celsius 79
8.2 Review Exercises 80
8.3 Review Questions 81
Chapter 9 Manipulating Numbers 83
9.1 Introduction 83
9.2 Useful Methods 83
9.3 Type Casting 86
9.4 Review Questions: True/False 87
9.5 Review Exercises 87
9.6 Review Questions 88
Chapter 10 Manipulating Strings 89
10.1 Introduction 89
10.2 Retrieving Individual Characters from a String 89
Exercise 10.2.1 — Displaying a String Backward 91
10.3 Useful Methods and Properties 91
Exercise 10.3.1 — Creating a Login ID 94
Exercise 10.3.2 — Switching the Order of Names 95
Exercise 10.3.3 — Creating a Random Word 96
10.4 Review Questions: True/False 97
10.5 Review Questions: Multiple Choice 98
10.6 Review Exercises 99
10.7 Review Questions 100
Chapter 11 Making Questions 101
11.1 Introduction 101
11.2 How to Write Simple Questions 101
Exercise 11.2.1 — Filling in the Table 102
11.3 Logical Operators and Complex Questions 103
11.4 What is the Order of Precedence of Logical Operators? 105
11.5 What is the Order of Precedence of Arithmetic, Comparison, and Logical Operators? 106
Exercise 11.5.1 — Filling in the Truth Table 106
11.6 Converting English Sentences to Boolean Expressions 107
11.7 Review Questions: True/False 109
11.8 Review Questions: Multiple Choice 110
11.9 Review Exercises 111

11.10 Review Questions 112

Chapter 12 Asking Questions - The `if` Structure 113

12.1 The `if` Structure 113
Exercise 12.1.1 — Find Out What is Displayed 115
Exercise 12.1.2 — Are you Allowed to Drive a Car? 116
Exercise 12.1.3 — Finding Minimum and Maximum Values with `if` Structures 116
Exercise 12.1.4 — Finding the Name of the Heaviest Person 118
12.2 Review Questions: True/False 119
12.3 Review Questions: Multiple Choice 119
12.4 Review Exercises 120
12.5 Review Questions 122

Chapter 13 Asking Questions - The `if-else` Structure 123

13.1 The `if-else` Structure 123
Exercise 13.1.1 — Finding the Output Message 124
Exercise 13.1.2 — Who is the Greatest? 124
Exercise 13.1.3 — Converting Gallons to Liters, and Vice Versa 126
13.2 Review Questions: True/False 126
13.3 Review Questions: Multiple Choice 127
13.4 Review Exercises 128

Chapter 14 Asking Questions - The `if-else-if` Structure 131

14.1 The `if-else-if` Structure 131
Exercise 14.1.1 — Find Out What is Displayed 132
Exercise 14.1.2 — Counting the Digits 134
Exercise 14.1.3 — The Days of the Week 135
Exercise 14.1.4 — Where is the Tollkeeper? 136
14.2 Review Questions: True/False 137
14.3 Review Exercises 137

Chapter 15 Asking Questions - Nested Structures 143

15.1 Nested Decision Structures 143
Exercise 15.1.1 — Find Out What is Displayed 144
Exercise 15.1.2 — Positive, Negative, or Zero? 145
Exercise 15.1.3 — The Most Scientific Calculator Ever! 146
15.2 Review Questions: True/False 147
15.3 Review Exercises 147
15.4 Review Questions 149

Chapter 16 Doing Loops 151

16.1 What is a Loop Structure? 151

16.2 From Sequence to Loop Structure 151
16.3 Review Questions: True/False 153

Chapter 17 Doing Loops - The while and the do-while Structures 155

17.1 The while Structure 155
Exercise 17.1.1 — Counting the Total Number of Iterations 156
Exercise 17.1.2 — Finding the Sum of Four Numbers 156
Exercise 17.1.3 — Finding the Sum of Positive Numbers 158
Exercise 17.1.4 — Finding the Sum of N Numbers 158
Exercise 17.1.5 — Finding the Sum of an Unknown Quantity of Numbers 159
Exercise 17.1.6 — Finding the Product of Five Numbers 160
17.2 The do-while Structure 161
Exercise 17.2.1 — Counting the Total Number of Iterations 162
Exercise 17.2.2 — Counting 2-digit Numbers 163
Exercise 17.2.3 — How Many Numbers Fit in a Sum 163
17.3 Review Questions: True/False 164
17.4 Review Questions: Multiple Choice 165
17.5 Review Exercises 168

Chapter 18 Doing Loops - The for and the foreach Structures 171

18.1 The for Structure 171
Exercise 18.1.1 — Find Out What is Displayed 173
Exercise 18.1.2 — Find Out What is Displayed 173
Exercise 18.1.3 — Finding the Sum of Four Numbers 174
Exercise 18.1.4 — Finding the Average Value of N Numbers 175
18.2 The foreach Structure 176
18.3 Review Questions: True/False 177
18.4 Review Questions: Multiple Choice 178
18.5 Review Exercises 180

Chapter 19 Doing Loops - Nested Structures 183

19.1 Nested Loop Structures 183
Exercise 19.1.1 — Counting the Total Number of Iterations. 184
Exercise 19.1.2 — Find Out What is Displayed 185
19.2 Review Questions: True/False 185
19.3 Review Questions: Multiple Choice 186
19.4 Review Exercises 188

Chapter 20 Tips and Tricks with Loop Structures 191

20.1 Introduction 191
20.2 Choosing a Loop Structure 191
20.3 The “Ultimate” Rule 192

20.4 Breaking Out of a Loop 195
20.5 Endless Loops and How to Avoid Them 196
20.6 The "From Inner to Outer" Method 197
20.7 Review Questions: True/False 199
20.8 Review Questions: Multiple Choice 199
20.9 Review Exercises 200

Chapter 21 More Exercises with Loop Structures 203

21.1 Exercises of a General Nature with Loop Structures 203
Exercise 21.1.1 — Finding the Sum of 1 + 2 + 3 + … + 100 203
Exercise 21.1.2 — Finding the Product of 2 × 4 × 6 × 8 × 10 204
Exercise 21.1.3 — Finding the Average Value of Positive Numbers 205
Exercise 21.1.4 — Counting the Numbers According to Which is Greater 206
Exercise 21.1.5 — Counting the Numbers According to Their Digits 207
Exercise 21.1.6 — Iterating as Many Times as the User Wants 208
Exercise 21.1.7 — Finding Minimum Value with Loop Structures 209
Exercise 21.1.8 — Fahrenheit to Kelvin, from 0 to 100 210
Exercise 21.1.9 — Rice on a Chessboard 211
Exercise 21.1.10 — Game – Find the Secret Number 212
21.2 Review Exercises 213

Chapter 22 Turtle Graphics 217

22.1 Introduction 217
22.2 The x-y Plane 217
22.3 Where is the Turtle? 219
22.4 Moving Forward and Backward 223
22.5 Turning Left and Right 224
Exercise 22.5.1 — Drawing a Rectangle 226
Exercise 22.5.2 — Drawing a Rectangle of Custom Size 227
22.6 Set the Orientation to a Specified Angle 227
22.7 Setting the Delay 228
22.8 Changing Pen's Color and Size 229
22.9 Pulling Turtle's Pen Up or Down 230
Exercise 22.9.1 — Drawing a House 231
22.10 Moving a Turtle Directly to a Specified Position 233
22.11 Using Decision and Loop Structures with Turtles 234
Exercise 22.11.1 — Drawing Squares of Different Sizes 237
Exercise 22.11.2 — Drawing Houses of Different Sizes 238
Exercise 22.11.3 — Drawing Polygons 240
Exercise 22.11.4 — Drawing a Star 241
Exercise 22.11.5 — Drawing Random Stars at Random Positions 242

Exercise 22.11.6 — Using Decision Structures to Draw Stars 244
22.12 Review Exercises 245

Chapter 23 Data Structures in C# 249

23.1 Introduction to Arrays 249
23.2 What is an Array? 250
Exercise 23.2.1 — Designing an Array 250
Exercise 23.2.2 — Designing Arrays 251
23.3 Creating Arrays in C# 252
23.4 How to Get a Value from an Array 253
Exercise 23.4.1 — Find What is Displayed 253
Exercise 23.4.2 — Using a Non-Existing Index in Arrays 254
23.5 How to Alter the Value of an Array Element 255
23.6 How to Iterate Through an Array 255
Exercise 23.6.1 — Finding the Sum 257
23.7 How to Add User-Entered Values to an Array 258
Exercise 23.7.1 — Displaying Words in Reverse Order 258
Exercise 23.7.2 — Displaying Positive Numbers in Reverse Order 259
Exercise 23.7.3 — Finding the Sum 260
Exercise 23.7.4 — Finding the Average Value 261
Exercise 23.7.5 — Displaying Reals Only 262
Exercise 23.7.6 — Displaying Odd Indexes Only 262
23.8 What is a Dictionary? 263
23.9 Creating Dictionaries in C# 263
23.10 How to Get a Value from a Dictionary 265
Exercise 23.10.1 — Using a Non-Existing Key in Dictionaries 265
23.11 How to Alter the Value of a Dictionary Element 266
Exercise 23.11.1 — Assigning a Value to a Non-Existing Key 266
23.12 How to Iterate Through a Dictionary 267
23.13 Useful Methods and Properties 268
23.14 Review Questions: True/False 270
23.15 Review Questions: Multiple Choice 274
23.16 Review Exercises 277
23.17 Review Questions 280

Chapter 24 More Exercises with Arrays 281

24.1 Simple Exercises with Arrays 281
Exercise 24.1.1 — Creating an Array with the Greatest Values 281
Exercise 24.1.2 — On Which Days Was There a Possibility of Snow? 282
Exercise 24.1.3 — Was There Any Possibility of Snow? 282
24.2 How to Use More Than One Arrays in a Program 284

Exercise 24.2.1 — Finding the Average Value 285
Exercise 24.2.2 — Using an Array Along with a Dictionary 286
24.3 Finding Minimum and Maximum Values in Arrays 287
Exercise 24.3.1 — Which Depth is the Greatest? 287
Exercise 24.3.2 — Which Lake is the Deepest? 288
Exercise 24.3.3 — Which Lake, in Which Country, Having Which Average Area, is the Deepest? 289
Exercise 24.3.4 — Which Students are the Shortest? 292
24.4 Searching Elements in Data Structures 293
Exercise 24.4.1 — Searching in an Array That May Contain the Same Value Multiple Times 293
Exercise 24.4.2 — Display the Last Names of All Those People Who Have the Same First Name 294
Exercise 24.4.3 — Searching in an Array that Contains Unique Values 295
Exercise 24.4.4 — Searching for a Given Social Security Number 295
24.5 Review Questions: True/False 296
24.6 Review Exercises 297

Chapter 25 Introduction to Subprograms 299

25.1 What is a Subprogram? 299
25.2 What is Procedural Programming? 299
25.3 What is Modular Programming? 300
25.4 Review Questions: True/False 301
25.5 Review Questions 301

Chapter 26 User-Defined Subprograms 303

26.1 Subprograms that Return Values 303
26.2 How to Call a Method that Returns Values 304
26.3 Subprograms that Return no Values 307
26.4 How to Call a Method that Returns no Values 308
26.5 Formal and Actual Arguments 309
26.6 How Does a Subprogram Execute? 310
26.7 Can Two Subprograms Use Variables of the Same Name? 311
26.8 Can a Subprogram Call Another Subprogram? 312
26.9 Default Argument Values and Named Arguments 313
26.10 The Scope of a Variable 314
26.11 Review Questions: True/False 316
26.12 Review Exercises 318
26.13 Review Questions 319

Chapter 27 More Exercises with Subprograms 321

27.1 Some More Exercises for Extra Practice 321
Exercise 27.1.1 — Back to Basics – Calculating the Sum of Two Numbers 321
Exercise 27.1.2 — Calculating the Sum of Two Numbers Using Fewer Lines of Code! 321
Exercise 27.1.3 — A Simple Currency Converter 322

Exercise 27.1.4 — A More Complete Currency Converter 323
Exercise 27.1.5 — Finding the Average Values of Positive Integers 325
Exercise 27.1.6 — Roll, Roll, Roll the… Dice! 326
27.2 Review Exercises 328

Chapter 28 Object-Oriented Programming 331

28.1 What is Object-Oriented Programming? 331
28.2 Classes and Objects in C# 332
28.3 The Constructor and the Keyword `this` 334
28.4 Passing Initial Values to the Constructor 336
Exercise 28.4.1 — Historical Events 337
28.5 Getter and Setter Methods vs Properties 339
Exercise 28.5.1 — The Roman Numerals 343
28.6 Can a Method Call Another Method of the Same Class? 346
Exercise 28.6.1 — Doing Math 347
28.7 Class Inheritance 348
28.8 Review Questions: True/False 353
28.9 Review Exercises 354
28.10 Review Questions 358

Some Final Words from the Authors 359

Index 361