Contents

Foreword, by Guido van Rossum ix
Preface x

Chapter 1 Computers and Programs 1

1.1 The Universal Machine 1
1.2 Program Power 3
1.3 What Is Computer Science? 3
1.4 Hardware Basics 5
1.5 Programming Languages 6
1.6 The Magic of Python 9
1.7 Inside a Python Program 15
1.8 Chaos and Computers 18
1.9 Chapter Summary 20
1.10 Exercises 21

Chapter 2 Writing Simple Programs 27

2.1 The Software Development Process 27
2.2 Example Program: Temperature Converter 28
2.3 Elements of Programs 31
2.3.1 Names 31
2.3.2 Expressions 32
2.4 Output Statements 34
2.5 Assignment Statements 36
2.5.1 Simple Assignment 37
2.5.2 Assigning Input 39
2.5.3 Simultaneous Assignment 41
2.6 Definite Loops 43

2.7 Example Program: Future Value 47
2.8 Chapter Summary 50
2.9 Exercises 51

Chapter 3 Computing with Numbers 57
3.1 Numeric Data Types 57
3.2 Type Conversions and Rounding 62
3.3 Using the Math Library 65
3.4 Accumulating Results: Factorials 68
3.5 Limitations of Computer Arithmetic 71
3.6 Chapter Summary 75
3.7 Exercises 76

Chapter 4 Objects and Graphics 83
4.1 Overview 83
4.2 The Object of Objects 84
4.3 Simple Graphics Programming 85
4.4 Using Graphical Objects 91
4.5 Graphing Future Value 96
4.6 Choosing Coordinates 103
4.7 Interactive Graphics 107
4.7.1 Getting Mouse Clicks 107
4.7.2 Handling Textual Input 109
4.8 Graphics Module Reference 112
4.8.1 GraphWin Objects 113
4.8.2 Graphics Objects 115
4.8.3 Entry Objects 119
4.8.4 Displaying Images 120
4.8.5 Generating Colors 121
4.8.6 Controlling Display Updates (Advanced) 121
4.9 Chapter Summary 122
4.10 Exercises 123

Chapter 5 Sequences: Strings, Lists, and Files 129
5.1 The String Data Type 129
5.2 Simple String Processing 133
5.3 Lists as Sequences 136
5.4 String Representation and Message Encoding 139
5.4.1 String Representation 139
5.4.2 Programming an Encoder 141
5.5 String Methods 142
5.5.1 Programming a Decoder 142
5.5.2 More String Methods 146
5.6 Lists Have Methods, Too 147
5.7 From Encoding to Encryption 150

5.8 Input/Output as String Manipulation 151
5.8.1 Example Application: Date Conversion 151
5.8.2 String Formatting 154
5.8.3 Better Change Counter 157
5.9 File Processing 158
5.9.1 Multi-line Strings 158
5.9.2 File Processing 159
5.9.3 Example Program: Batch Usernames 163
5.9.4 File Dialogs (Optional) 164
5.10 Chapter Summary 167
5.11 Exercises 168

Chapter 6 Defining Functions 175
6.1 The Function of Functions 175
6.2 Functions, Informally 177
6.3 Future Value with a Function 181
6.4 Functions and Parameters: The Exciting Details 183
6.5 Functions That Return Values 187
6.6 Functions that Modify Parameters 193
6.7 Functions and Program Structure 199
6.8 Chapter Summary 202
6.9 Exercises 203

Chapter 7 Decision Structures 209
7.1 Simple Decisions 209
7.1.1 Example: Temperature Warnings 210
7.1.2 Forming Simple Conditions 212
7.1.3 Example: Conditional Program Execution 214
7.2 Two-Way Decisions 216
7.3 Multi-Way Decisions 220
7.4 Exception Handling 223
7.5 Study in Design: Max of Three 227
7.5.1 Strategy 1: Compare Each to All 228
7.5.2 Strategy 2: Decision Tree 230
7.5.3 Strategy 3: Sequential Processing 231
7.5.4 Strategy 4: Use Python 234
7.5.5 Some Lessons 234
7.6 Chapter Summary 235
7.7 Exercises 236

Chapter 8 Loop Structures and Booleans 243
8.1 For Loops: A Quick Review 243
8.2 Indefinite Loops 245
8.3 Common Loop Patterns 247
8.3.1 Interactive Loops 247
8.3.2 Sentinel Loops 249

8.3.3 File Loops 252
8.3.4 Nested Loops 254
8.4 Computing with Booleans 256
8.4.1 Boolean Operators 256
8.4.2 Boolean Algebra 260
8.5 Other Common Structures 262
8.5.1 Post-test Loop 262
8.5.2 Loop and a Half 264
8.5.3 Boolean Expressions as Decisions 266
8.6 Example: A Simple Event Loop 269
8.7 Chapter Summary 275
8.8 Exercises 277

Chapter 9 Simulation and Design 283

9.1 Simulating Racquetball 283
9.1.1 A Simulation Problem 284
9.1.2 Analysis and Specification 284
9.2 Pseudo-random Numbers 286
9.3 Top-Down Design 288
9.3.1 Top-Level Design 289
9.3.2 Separation of Concerns 291
9.3.3 Second-Level Design 291
9.3.4 Designing simNGames 293
9.3.5 Third-Level Design 295
9.3.6 Finishing Up 298
9.3.7 Summary of the Design Process 300
9.4 Bottom-Up Implementation 301
9.4.1 Unit Testing 301
9.4.2 Simulation Results 303
9.5 Other Design Techniques 304
9.5.1 Prototyping and Spiral Development 304
9.5.2 The Art of Design 306
9.6 Chapter Summary 306
9.7 Exercises 307

Chapter 10 Defining Classes 313

10.1 Quick Review of Objects 313
10.2 Example Program: Cannonball 314
10.2.1 Program Specification 314
10.2.2 Designing the Program 315
10.2.3 Modularizing the Program 319
10.3 Defining New Classes 321
10.3.1 Example: Multi-sided Dice 321
10.3.2 Example: The Projectile Class 325
10.4 Data Processing with Class 327
10.5 Objects and Encapsulation 331

10.5.1 Encapsulating Useful Abstractions 331
10.5.2 Putting Classes in Modules 333
10.5.3 Module Documentation 333
10.5.4 Working with Multiple Modules 335
10.6 Widgets 337
10.6.1 Example Program: Dice Roller 337
10.6.2 Building Buttons 338
10.6.3 Building Dice 342
10.6.4 The Main Program 345
10.7 Animated Cannonball 346
10.7.1 Drawing the Animation Window 347
10.7.2 Creating a ShotTracker 348
10.7.3 Creating an Input Dialog 350
10.7.4 The Main Event Loop 353
10.8 Chapter Summary 355
10.9 Exercises 356

Chapter 11 Data Collections 363

11.1 Example Problem: Simple Statistics 363
11.2 Applying Lists 365
11.2.1 Lists and Arrays 366
11.2.2 List Operations 367
11.2.3 Statistics with Lists 370
11.3 Lists of Records 375
11.4 Designing with Lists and Classes 379
11.5 Case Study: Python Calculator 385
11.5.1 A Calculator as an Object 385
11.5.2 Constructing the Interface 385
11.5.3 Processing Buttons 388
11.6 Case Study: Better Cannonball Animation 392
11.6.1 Creating a Launcher 393
11.6.2 Tracking Multiple Shots 396
11.7 Non-sequential Collections 401
11.7.1 Dictionary Basics 401
11.7.2 Dictionary Operations 402
11.7.3 Example Program: Word Frequency 404
11.8 Chapter Summary 409
11.9 Exercises 410

Chapter 12 Object-Oriented Design 419

12.1 The Process of OOD 419
12.2 Case Study: Racquetball Simulation 422
12.2.1 Candidate Objects and Methods 422
12.2.2 Implementing SimStats 424
12.2.3 Implementing RBallGame 426
12.2.4 Implementing Player 429

12.2.5 The Complete Program 430
12.3 Case Study: Dice Poker 433
12.3.1 Program Specification 433
12.3.2 Identifying Candidate Objects 434
12.3.3 Implementing the Model 436
12.3.4 A Text-Based UI 440
12.3.5 Developing a GUI 443
12.4 OO Concepts 451
12.4.1 Encapsulation 452
12.4.2 Polymorphism 453
12.4.3 Inheritance 453
12.5 Chapter Summary 455
12.6 Exercises 456

Chapter 13 Algorithm Design and Recursion 459
13.1 Searching 460
13.1.1 A Simple Searching Problem 460
13.1.2 Strategy 1: Linear Search 461
13.1.3 Strategy 2: Binary Search 462
13.1.4 Comparing Algorithms 463
13.2 Recursive Problem Solving 465
13.2.1 Recursive Definitions 466
13.2.2 Recursive Functions 468
13.2.3 Example: String Reversal 469
13.2.4 Example: Anagrams 471
13.2.5 Example: Fast Exponentiation 472
13.2.6 Example: Binary Search 473
13.2.7 Recursion vs. Iteration 474
13.3 Sorting Algorithms 477
13.3.1 Naive Sorting: Selection Sort 477
13.3.2 Divide and Conquer: Merge Sort 479
13.3.3 Comparing Sorts 481
13.4 Hard Problems 484
13.4.1 Tower of Hanoi 484
13.4.2 The Halting Problem 489
13.4.3 Conclusion 492
13.5 Chapter Summary 493
13.6 Exercises 494

Appendix A Python Quick Reference 503
Appendix C Glossary 513
Index 525