Introduction to Computer Science Using Python: A Computational Problem-Solving Focus

Charles Dierbach
2 Data and Expressions

MOTIVATION 39
FUNDAMENTAL CONCEPTS 40

2.1 Literals 40
  2.1.1 What Is a Literal? 40
  2.1.2 Numeric Literals 40
  2.1.3 String Literals 44
  2.1.4 Control Characters 46
  2.1.5 String Formatting 47
  2.1.6 Implicit and Explicit Line Joining 48
  2.1.7 Let's Apply It—“Hello World Unicode Encoding” 48
  Self-Test Questions 49

2.2 Variables and Identifiers 50
  2.2.1 What Is a Variable? 50
  2.2.2 Variable Assignment and Keyboard Input 52
  2.2.3 What Is an Identifier? 53
  2.2.4 Keywords and Other Predefined Identifiers in Python 54
  2.2.5 Let’s Apply It—“Restaurant Tab Calculation” 55
  Self-Test Questions 56

2.3 Operators 57
  2.3.1 What Is an Operator? 57
  2.3.2 Arithmetic Operators 57
  2.3.3 Let’s Apply It—“Your Place in the Universe” 59
  Self-Test Questions 60

2.4 Expressions and Data Types 61
  2.4.1 What Is an Expression? 61
  2.4.2 Operator Precedence 61
  2.4.3 Operator Associativity 63
2.4.4 What Is a Data Type?  64
2.4.5 Mixed-Type Expressions  64
2.4.6 Let’s Apply It—“Temperature Conversion Program”  65
Self-Test Questions  66

COMPUTATIONAL PROBLEM SOLVING  67
2.5 Age in Seconds Program  67
2.5.1 The Problem  67
2.5.2 Problem Analysis  67
2.5.3 Program Design  67
2.5.4 Program Implementation and Testing  69

Chapter Summary  74
Chapter Exercises  74

3 Control Structures  79

MOTIVATION  80
FUNDAMENTAL CONCEPTS  80
3.1 What Is a Control Structure?  80
3.2 Boolean Expressions (Conditions)  81
3.2.1 Relational Operators  81
3.2.2 Membership Operators  82
3.2.3 Boolean Operators  83
3.2.4 Operator Precedence and Boolean Expressions  85
3.2.5 Short-Circuit (Lazy) Evaluation  86
3.2.6 Logically Equivalent Boolean Expressions  87
Self-Test Questions  88
3.3 Selection Control  89
3.3.1 If Statement  89
3.3.2 Indentation in Python  90
3.3.3 Multi-Way Selection  91
3.3.4 Let’s Apply It—Number of Days in Month Program  94
Self-Test Questions  96
3.4 Iterative Control  96
3.4.1 While Statement  97
3.4.2 Input Error Checking  98
3.4.3 Infinite loops  99
3.4.4 Definite vs. Indefinite Loops  100
3.4.5 Boolean Flags and Indefinite Loops  100
3.4.6 Let’s Apply It—Coin Change Exercise Program  101
Self-Test Questions  104

COMPUTATIONAL PROBLEM SOLVING  104
3.5 Calendar Month Program  104
3.5.1 The Problem  104
MOTIVATION 126
FUNDAMENTAL CONCEPTS 127
4.1 List Structures 127
  4.1.1 What Is a List? 127
  4.1.2 Common List Operations 127
  4.1.3 List Traversal 128
  Self-Test Questions 129
4.2 Lists (Sequences) in Python 130
  4.2.1 Python List Type 130
  4.2.2 Tuples 131
  4.2.3 Sequences 132
  4.2.4 Nested Lists 134
  4.2.5 Let's Apply It—A Chinese Zodiac Program 135
  Self-Test Questions 137
4.3 Iterating Over Lists (Sequences) in Python 137
  4.3.1 For Loops 137
  4.3.2 The Built-in range Function 138
  4.3.3 Iterating Over List Elements vs. List Index Values 139
  4.3.4 While Loops and Lists (Sequences) 140
  4.3.5 Let's Apply It—Password Encryption/Decryption Program 141
  Self-Test Questions 144
4.4 More on Python Lists 144
  4.4.1 Assigning and Copying Lists 144
  4.4.2 List Comprehensions 146

COMPUTATIONAL PROBLEM SOLVING 147
4.5 Calendar Year Program 147
  4.5.1 The Problem 147
  4.5.2 Problem Analysis 147
  4.5.3 Program Design 148
  4.5.4 Program Implementation and Testing 149
Chapter Summary 161
Chapter Exercises 162
Python Programming Exercises 164
Program Modification Problems 164
Program Development Problems 165
6.2.6 Let's Apply It—Bouncing Balls Program 226
Self-Test Questions 229

COMPUTATIONAL PROBLEM SOLVING 229
6.3 Horse Race Simulation Program 229
   6.3.1 The Problem 230
   6.3.2 Problem Analysis 230
   6.3.3 Program Design 231
   6.3.4 Program Implementation and Testing 231
Chapter Summary 243
Chapter Exercises 243
Python Programming Exercises 244
Program Modification Problems 245
Program Development Problems 246

Chapter 7
Modular Design

MOTIVATION 248
FUNDAMENTAL CONCEPTS 248
7.1 Modules 248
   7.1.1 What Is a Module? 248
   7.1.2 Module Specification 249
      Self-Test Questions 251
7.2 Top-Down Design 251
   7.2.1 Developing a Modular Design of the Calendar Year Program 251
   7.2.2 Specification of the Calendar Year Program Modules 252
      Self-Test Questions 255
7.3 Python Modules 255
   7.3.1 What Is a Python Module? 255
   7.3.2 Modules and Namespaces 256
   7.3.3 Importing Modules 257
   7.3.4 Module Loading and Execution 260
   7.3.5 Local, Global, and Built-in Namespaces in Python 262
   7.3.6 A Programmer-Defined Stack Module 264
   7.3.7 Let's Apply It—A Palindrome Checker Program 267
      Self-Test Questions 268

COMPUTATIONAL PROBLEM SOLVING 269
7.4 Calendar Year Program (function version) 269
   7.4.1 The Problem 269
   7.4.2 Problem Analysis 269
   7.4.3 Program Design 269
   7.4.4 Program Implementation and Testing 269
Chapter Summary 284
Chapter Exercises 284
Python Programming Exercises 286
Program Modification Problems 287
Program Development Problems 287
10 Object-Oriented Programming

MOTIVATION 384
FUNDAMENTAL CONCEPTS 384
10.1 What Is Object-Oriented Programming? 384
  10.1.1 What Is a Class? 385
  10.1.2 Three Fundamental Features of Object-Oriented Programming 385
10.2 Encapsulation 386
  10.2.1 What Is Encapsulation? 386
  10.2.2 Defining Classes in Python 387
  10.2.3 Let's Apply It—A Recipe Conversion Program 394
  Self-Test Questions 399
10.3 Inheritance 400
  10.3.1 What Is Inheritance? 400
  10.3.2 Subtypes 401
  10.3.3 Defining Subclasses in Python 402
  10.3.4 Let's Apply It—A Mixed Fraction Class 407
  Self-Test Questions 411
10.4 Polymorphism 411
  10.4.1 What Is Polymorphism? 411
  10.4.2 The Use of Polymorphism 414
  Self-Test Questions 417
10.5 Object-Oriented Design Using UML 417
  10.5.1 What Is UML? 417
  10.5.2 UML Class Diagrams 418
  Self-Test Questions 422

COMPUTATIONAL PROBLEM SOLVING 423
10.6 Vehicle Rental Agency Program 423
  10.6.1 The Problem 423
12.2.4 The Development of Memory Electronic Circuits (1919) 495
12.2.5 The Development of Electronic Digital Logic Circuits (1937) 495
12.2.6 "The Father of Information Theory"—Claude Shannon (1948) 496

FIRST-GENERATION COMPUTERS (1940s–mid-1950s) 496
12.3 The Early Groundbreakers 496
  12.3.1 The Z3—The First Programmable Computer (1941) 496
  12.3.2 The Mark I—First Computer Project in the United States (1937–1943) 497
  12.3.3 The ABC—The First Fully Electronic Computing Device (1942) 498
  12.3.4 Colossus—A Special-Purpose Electronic Computer (1943) 499
  12.3.5 ENIAC—The First Fully Electronic Programmable Computer 500
  12.3.6 EDVAC/ACE—The First Stored Program Computers (1950) 501
  12.3.7 Whirlwind—The First Real-Time Computer (1951) 502

12.4 The First Commercially Available Computers 503
  12.4.1 The Struggles of the Eckert-Mauchly Computer Corporation (1950) 503
  12.4.2 The LEO Computer of the J. Lyons and Company (1951) 504

SECOND-GENERATION COMPUTERS (mid-1950s to mid-1960s) 505
12.5 Transistorized Computers 505
  12.5.1 The Development of the Transistor (1947) 505
  12.5.2 The First Transistor Computer (1953) 506

12.6 The Development of High-Level Programming Languages 506
  12.6.1 The Development of Assembly Language (early 1950s) 506
  12.6.2 The First High-Level Programming Languages (mid-1950s) 507
  12.6.3 The First "Program Bug" (1947) 508

THIRD-GENERATION COMPUTERS (mid-1960s to early 1970s) 508
12.7 The Development of the Integrated Circuit (1958) 508
  12.7.1 The Catalyst for Integrated Circuit Advancements (1960s) 509
  12.7.2 The Development of the Microprocessor (1971) 511

12.8 Mainframes, Minicomputers, and Supercomputers 512
  12.8.1 The Establishment of the Mainframe Computer (1962) 512
  12.8.2 The Development of the Minicomputer (1963) 513
  12.8.3 The Development of the UNIX Operating System (1969) 513
  12.8.4 The Development of Graphical User Interfaces (early 1960s) 514
  12.8.5 The Development of the Supercomputer (1972) 515

FOURTH-GENERATION COMPUTERS (early 1970s to the Present) 515
12.9 The Rise of the Microprocessor 515
  12.9.1 The First Commercially Available Microprocessor (1971) 515
  12.9.2 The First Commercially Available Microcomputer Kit (1975) 516

12.10 The Dawn of Personal Computing 516
  12.10.1 The Beginnings of Microsoft (1975) 516
  12.10.2 The Apple II (1977) 517
  12.10.3 IBM’s Entry into the Microcomputer Market (1981) 517
  12.10.4 Society Embraces the Personal Computer (1983) 518
  12.10.5 The Development of Graphical User Interfaces (GUIs) 518
  12.10.6 The Development of the C++ Programming Language 519
THE DEVELOPMENT OF COMPUTER NETWORKS  520
12.11  The Development of Wide Area Networks  520
   12.11.1  The Idea of Packet-Switched Networks (early 1960s)  520
   12.11.2  The First Packet-Switched Network: ARPANET (1969)  520
12.12  The Development of Local Area Networks (LANs)  521
   12.12.1  The Need for Local Area Networks  521
   12.12.2  The Development of Ethernet (1980)  521
12.13  The Development of the Internet and World Wide Web  522
   12.13.1  The Realization of the Need for “Internetworking”  522
   12.13.2  The Development of the TCP/IP Internetworking Protocol (1973)  522
   12.13.3  The Development of the World Wide Web (1990)  522
   12.13.4  The Development of the Java Programming Language (1995)  523

Appendix  525
Index  569