

Table of Contents

| | |
|--|------|
| Introduction | xiii |
| Part I Basic Concepts | 1 |
| Chapter 1 Integer Arithmetic | 3 |
| 1.1 Example Program | 3 |
| 1.2 Computer Program | 4 |
| 1.3 Documentation | 5 |
| 1.4 Input | 6 |
| 1.5 Assignment Statement | 7 |
| 1.5.1 Basics of assignment statements | 8 |
| 1.5.2 Classification of errors | 9 |
| 1.5.3 Semantics of assignment statements | 9 |
| 1.5.4 Style guide for naming variables | 12 |
| 1.6 Built-in Types | 12 |
| 1.7 Integer types | 13 |
| 1.7.1 Boolean type | 18 |
| 1.8 Selection | 21 |
| 1.8.1 Backus-Naur Form | 22 |
| 1.8.2 The if statement | 25 |
| 1.9 Output | 27 |
| 1.10 Key Points | 29 |
| 1.11 Assignments | 29 |
| Chapter 2 Floating-Point Arithmetic | 31 |
| 2.1 Example Program | 31 |
| 2.2 Floating-point Type | 33 |
| 2.2.1 Floating-point representation | 33 |
| 2.2.2 Rounding errors | 34 |
| 2.2.3 Floating-point literals | 35 |
| 2.2.4 The built-in function float | 36 |
| 2.2.5 Floating-point operators and built-ins | 37 |
| 2.2.6 Built-in functions | 40 |
| 2.2.7 Library functions | 41 |
| 2.3 Key Points | 42 |
| 2.4 Assignment | 42 |
| Chapter 3 Informal Approach to Iteration | 45 |
| 3.1 Example Programs | 45 |

| | |
|---|-----------|
| 3.2 Algorithms | 48 |
| 3.2.1 High-level descriptions of algorithms | 48 |
| 3.2.2 Components of iterative algorithms | 49 |
| 3.3 Iterative Algorithms in Python | 50 |
| 3.3.1 Digit sum | 51 |
| 3.3.2 Square root | 52 |
| 3.4 Symbolic Constants | 53 |
| 3.5 Key Points | 54 |
| 3.6 Assignment | 54 |
| Chapter 4 Strings | 57 |
| 4.1 Example Program | 57 |
| 4.2 String Literals | 58 |
| 4.2.1 ASCII | 59 |
| 4.2.2 Unicode | 59 |
| 4.3 Indexing | 61 |
| 4.4 Slicing | 62 |
| 4.5 Comparison | 63 |
| 4.6 Operators and Functions | 64 |
| 4.7 Key Points | 65 |
| 4.8 Assignment | 66 |
| Part II Structuring Concepts | 67 |
| Chapter 5 Functions | 69 |
| 5.1 Example Function | 69 |
| 5.2 Function Definition | 70 |
| 5.2.1 Function heading | 71 |
| 5.2.2 Documentation string | 72 |
| 5.2.3 Function body | 74 |
| 5.2.4 Return statement | 76 |
| 5.2.5 A note on structured programming | 78 |
| 5.3 Function Invocation | 79 |
| 5.4 Testing | 81 |
| 5.4.1 Outline of a test suite | 82 |
| 5.4.2 Realization of test suite | 83 |
| 5.5 Key Points | 84 |
| 5.6 Assignment | 84 |
| Chapter 6 Tuples | 87 |
| 6.1 Example Function | 87 |
| 6.2 Design by Example | 88 |
| 6.3 Semantics of Sequences | 90 |
| 6.4 Tuple Literals | 91 |

| | |
|----------------------------------|-----|
| 6.5 Tuple Assignment | 93 |
| 6.6 Inspection of Sequences | 95 |
| 6.6.1 Indexing | 95 |
| 6.6.2 Slicing | 96 |
| 6.7 Comparison | 98 |
| 6.8 Concatenation | 100 |
| 6.9 Variable Number of Arguments | 100 |
| 6.10 Key Points | 102 |
| 6.11 Assignment | 102 |
| Chapter 7 Lists | 105 |
| 7.1 Example Function | 105 |
| 7.2 Selection Sort | 106 |
| 7.3 Semantics of Lists | 107 |
| 7.4 List Literals | 110 |
| 7.5 Operations | 111 |
| 7.6 Import Statement | 115 |
| 7.7 For Statement | 118 |
| 7.8 Ranges | 120 |
| 7.9 Key Points | 122 |
| 7.10 Assignment | 122 |
| Chapter 8 Matrices | 123 |
| 8.1 Example Functions | 123 |
| 8.2 Semantics of Matrices | 125 |
| 8.2.1 Matrix manipulation | 126 |
| 8.2.2 Matrix check | 127 |
| 8.3 Nested Loops | 128 |
| 8.4 List Comprehension | 130 |
| 8.5 Key Points | 132 |
| 8.6 Assignment | 132 |
| Chapter 9 Sets | 135 |
| 9.1 Example Function | 135 |
| 9.2 Semantics of Sets | 137 |
| 9.2.1 Set initialization | 137 |
| 9.2.2 Hashing | 139 |
| 9.3 Set Manipulation | 141 |
| 9.3.1 Set inspection | 141 |
| 9.3.2 Set computations | 142 |
| 9.3.3 Set mutation | 144 |
| 9.4 Iteration over Sets | 147 |
| 9.5 Key Points | 149 |

| | |
|--|-----|
| 9.6 Assignment | 149 |
| Chapter 10 Dictionaries | 151 |
| 10.1 Example Code | 151 |
| 10.2 Semantics of Dictionaries | 152 |
| 10.3 Dictionary Initialization | 153 |
| 10.4 Dictionary Inspection | 155 |
| 10.5 Dictionary Mutation | 156 |
| 10.6 Iterating over Dictionaries | 158 |
| 10.7 Key Points | 160 |
| 10.8 Assignment | 161 |
| Chapter 11 Simple Recursion | 163 |
| 11.1 Example Function | 163 |
| 11.2 Recursion in Mathematics | 164 |
| 11.2.1 Recursive sets | 164 |
| 11.2.2 Recursive functions | 165 |
| 11.2.3 Fractals | 167 |
| 11.3 Recursion in Informatics | 168 |
| 11.3.1 Mathematical definition of factorials | 168 |
| 11.3.2 Recursive implementation of the factorial | 169 |
| 11.3.3 Execution of recursive functions | 169 |
| 11.4 Recursion versus Iteration | 172 |
| 11.4.1 Recursive version | 172 |
| 11.4.2 Iterative version | 174 |
| 11.5 Key Points | 175 |
| 11.6 Assignment | 175 |
| Chapter 12 Multiple Recursion | 177 |
| 12.1 Non-Tail Recursion | 177 |
| 12.2 Backtracking | 184 |
| 12.2.1 Eight queens problem | 184 |
| 12.2.2 General schemes | 190 |
| 12.3 Key Points | 193 |
| 12.4 Assignment | 194 |
| Chapter 13 Divide and Conquer | 197 |
| 13.1 Introduction | 197 |
| 13.2 Example: Elections | 198 |
| 13.3 Implementation | 199 |
| 13.3.1 Counting Votes | 200 |
| 13.3.2 Allocating Seats | 202 |
| 13.3.3 Assigning Candidates | 204 |
| 13.4 Key Points | 207 |

| | |
|--|------------|
| Part III Advanced Topics | 209 |
| Chapter 14 Loop Invariants | 211 |
| 14.1 Example Functions | 211 |
| 14.2 Semantics of Loop Invariants | 213 |
| 14.3 Algorithm Design | 217 |
| 14.3.1 Integer sum | 218 |
| 14.3.2 Number of occurrences | 219 |
| 14.3.3 Strong versus weak tests | 220 |
| 14.4 Correctness of Algorithms | 221 |
| 14.4.1 Base step | 222 |
| 14.4.2 Induction step | 223 |
| 14.4.3 Epilogue | 224 |
| 14.5 Termination | 225 |
| 14.5.1 Loop variant | 226 |
| 14.5.2 Termination proof | 227 |
| 14.6 Correctness of Selection Sort | 228 |
| 14.7 Correctness of Binary Search | 231 |
| 14.8 Dutch National Flag Problem | 233 |
| 14.8.1 A humble programmer's version | 233 |
| 14.8.2 Dijkstra's version | 235 |
| 14.8.3 Proof of correctness | 236 |
| 14.8.4 Proof of finiteness | 240 |
| 14.9 Key Points | 240 |
| 14.10 Assignment | 241 |
| Chapter 15 Complexity | 243 |
| 15.1 Efficiency | 243 |
| 15.2 Iterative Algorithms | 245 |
| 15.2.1 Examples from previous chapters | 246 |
| 15.2.2 Insertion sort | 250 |
| 15.3 Recursive Algorithms | 251 |
| 15.3.1 Examples from previous chapters | 252 |
| 15.3.2 Heapsort | 255 |
| 15.4 Key Points | 262 |
| 15.5 Assignment | 262 |
| Chapter 16 Dynamic Programming | 265 |
| 16.1 Problem Statement | 265 |
| 16.1.1 Fibonacci Numbers | 266 |
| 16.2 Memoization | 268 |
| 16.3 The Knapsack Problem | 269 |
| 16.3.1 Brute Force Approach | 270 |
| 16.3.2 Greedy Approach | 273 |
| 16.3.3 Dynamic Programming Approach | 275 |

| | |
|---|-----|
| 16.4 Key Points | 278 |
| 16.5 Assignment | 278 |
| Chapter 17 Functional Programming | 281 |
| 17.1 Declarative Programming | 281 |
| 17.2 Imperative versus Declarative | 282 |
| 17.2.1 Iterative solution | 282 |
| 17.2.2 Recursive solution | 284 |
| 17.2.3 Solution with set comprehension | 286 |
| 17.3 Functions as First-Class Citizens | 288 |
| 17.4 Lambda Functions | 292 |
| 17.4.1 Definition | 292 |
| 17.4.2 Closures | 295 |
| 17.5 Higher-Order Functions | 298 |
| 17.5.1 Filter | 298 |
| 17.5.2 Map | 300 |
| 17.5.3 List comprehension and set comprehension | 302 |
| 17.5.4 Self-defined higher-order functions | 303 |
| 17.6 Key Points | 308 |
| 17.7 Assignment | 308 |
| Chapter 18 Object-Oriented Programming | 311 |
| 18.1 The Object-Oriented Paradigm | 311 |
| 18.2 Procedural Code | 312 |
| 18.3 Class Definition | 316 |
| 18.3.1 Class heading | 316 |
| 18.3.2 Attributes | 318 |
| 18.3.3 Methods | 320 |
| 18.3.4 Constructor | 323 |
| 18.3.5 Encapsulation | 326 |
| 18.3.6 Additional methods | 330 |
| 18.4 Key Points | 339 |
| 18.5 Assignment | 339 |
| Chapter 19 Inheritance | 341 |
| 19.1 Basics of inheritance | 341 |
| 19.2 Superclasses | 342 |
| 19.3 Subclasses | 348 |
| 19.4 Extending Subclasses | 350 |
| 19.5 Method Overriding | 353 |
| 19.6 Key Points | 361 |
| 19.7 Assignment | 362 |

| | |
|---------------------------------------|------------|
| Solutions to Assignments | 363 |
| Chapter 1: Increment Time | 363 |
| Chapter 2: Quadratic Equation | 365 |
| Chapter 3: Mantissa | 368 |
| Chapter 4: Hexadecimal representation | 370 |
| Chapter 4: Palindrome | 373 |
| Chapter 5: Approximation of π | 376 |
| Chapter 6: Sorted Sequences | 378 |
| Chapter 7: Duplicates | 382 |
| Chapter 8: Encryption | 385 |
| Chapter 9: Set Cover | 386 |
| Chapter 10: Sparse Vectors | 388 |
| Chapter 11: Power | 392 |
| Chapter 11: Palindrome | 393 |
| Chapter 12: Quicksort | 395 |
| Chapter 12: Sudoku | 398 |
| Chapter 14: Longest Plateau | 400 |
| Chapter 15: Bubble Sort | 404 |
| Chapter 15: Merge Sort | 406 |
| Chapter 16: Shortest Route | 410 |
| Chapter 17: Suspicious Bank Accounts | 412 |
| Chapter 17: Flu Prediction | 413 |
| Chapter 18: Coordinates | 414 |
| Chapter 19: Junior Accounts | 420 |
| Index | 425 |