# Contents

**Preface** ix
**Notes for Teachers** xv

**0. Introduction** 1
0.0. Prerequisites . . . 2
0.1. Outline of Topics . . . 4
0.2. Dafny . . . 5
0.3. Other Languages . . . 6

## Part 0. Learning the Ropes

**1. Basics** 9
1.0. Methods . . . 9
1.1. Assert Statements . . . 10
1.2. Working with the Verifier . . . 11
1.3. Control Paths . . . 12
1.4. Method Contracts . . . 13
1.5. Functions . . . 17
1.6. Compiled versus Ghost . . . 19
1.7. Summary . . . 21
**2. Making It Formal** 25
2.0. Program State . . . 26
2.1. Floyd Logic . . . 28
2.2. Hoare Triples . . . 29
2.3. Strongest Postconditions and Weakest Preconditions . . . 32
2.4. $\mathcal{WP}$ and $\mathcal{SP}$ . . . 40
2.5. Conditional Control Flow . . . 41
2.6. Sequential Composition . . . 45
2.7. Method Calls and Postconditions . . . 46
2.8. Assert Statements . . . 50
2.9. Weakest Liberal Preconditions . . . 53
2.10. Method Calls with Preconditions . . . 55
2.11. Function Calls . . . 57

2.12. Partial Expressions . . . 58
2.13. Method Correctness . . . 60
2.14. Summary . . . 60
**3. Recursion and Termination** 63
3.0. The Endless Problem . . . 64
3.1. Avoiding Infinite Recursion . . . 66
3.2. Well-Founded Relations . . . 70
3.3. Lexicographic Tuples . . . 72
3.4. Default **`decreases`** in Dafny . . . 79
3.5. Summary . . . 80
**4. Inductive Datatypes** 83
4.0. Blue-Yellow Trees . . . 84
4.1. Matching on Datatypes . . . 85
4.2. Discriminators and Destructors . . . 86
4.3. Structural Inclusion . . . 88
4.4. Enumerations . . . 89
4.5. Type Parameters . . . 89
4.6. Abstract Syntax Trees for Expressions . . . 90
4.7. Summary . . . 93
**5. Lemmas and Proofs** 95
5.0. Declaring a Lemma . . . 96
5.1. Using a Lemma . . . 96
5.2. Proving a Lemma . . . 99
5.3. Back to Basics . . . 102
5.4. Proof Calculations . . . 106
5.5. Example: Reduce . . . 110
5.6. Example: Commutativity of Multiplication . . . 115
5.7. Example: Mirroring a Tree . . . 118
5.8. Example: Working on Abstract Syntax Trees . . . 122
5.9. Summary . . . 130

## Part 1. Functional Programs

**6. Lists** 137
6.0. List Definition . . . 137
6.1. `Length` . . . 138
6.2. Intrinsic versus Extrinsic Specifications . . . 139
6.3. `Take` and `Drop` . . . 142
6.4. `At` . . . 144
6.5. `Find` . . . 146
6.6. List Reversal . . . 147
6.7. Lemmas in Expressions . . . 151
6.8. Eliding Type Arguments . . . 157
6.9. Summary . . . 158

**7. Unary Numbers** 161
7.0. Basic Definitions . . . 162
7.1. Comparisons . . . 162
7.2. Addition and Subtraction . . . 165
7.3. Multiplication . . . 167
7.4. Division and Modulus . . . 167
7.5. Summary . . . 172
**8. Sorting** 175
8.0. Specification . . . 175
8.1. Insertion Sort . . . 179
8.2. Merge Sort . . . 181
8.3. Summary . . . 188
**9. Abstraction** 189
9.0. Grouping Declarations into Modules . . . 190
9.1. Module Imports . . . 190
9.2. Export Sets . . . 191
9.3. Modular Specification of a Queue . . . 194
9.4. Equality-Supporting Types . . . 201
9.5. Summary . . . 204
**10. Data-Structure Invariants** 207
10.0. Priority-Queue Specification . . . 208
10.1. Designing the Data Structure . . . 210
10.2. Implementation . . . 212
10.3. Making Intrinsic from Extrinsic . . . 224
10.4. Summary . . . 229

## Part 2. Imperative Programs

**11. Loops** 235
11.0. Loop Specifications . . . 235
11.1. Loop Implementations . . . 241
11.2. Loop Termination . . . 247
11.3. Summarizing the Loop Rule . . . 250
11.4. Integer Square Root . . . 252
11.5. Summary . . . 255
**12. Recursive Specifications, Iterative Programs** 257
12.0. Iterative Fibonacci . . . 257
12.1. Fibonacci Squared . . . 260
12.2. Powers of 2 . . . 264
12.3. Sums . . . 267
12.4. Summary . . . 272
**13. Arrays and Searching** 275
13.0. About Arrays . . . 275
13.1. Linear Search . . . 280

13.2. Binary Search . . . 288
13.3. Minimum . . . 292
13.4. Coincidence Count . . . 294
13.5. Slope Search . . . 301
13.6. Canyon Search . . . 304
13.7. Majority Vote . . . 309
13.8. Summary . . . 318
**14. Modifying Arrays** 321
14.0. Simple Frames . . . 321
14.1. Basic Array Modification . . . 326
14.2. Summary . . . 336
**15. In-situ Sorting** 337
15.0. Dutch National Flag . . . 337
15.1. Selection Sort . . . 341
15.2. Quicksort . . . 343
15.3. Summary . . . 347
**16. Objects** 351
16.0. Checksums . . . 352
16.1. Tokenizer . . . 359
16.2. Simple Aggregate Objects . . . 364
16.3. Full Aggregate Objects . . . 374
16.4. Summary . . . 382
**17. Dynamic Heap Data Structures** 387
17.0. Lazily Initialized Arrays . . . 387
17.1. Extensible Array . . . 396
17.2. Binary Search Tree for a Map . . . 403
17.3. Iterator for the Map . . . 413
17.4. Summary . . . 423

**A. Dafny Syntax Cheat Sheet** 427
**B. Boolean Algebra** 433
B.0. Boolean Values and Negation . . . 433
B.1. Conjunction . . . 433
B.2. Predicates and Well-Definedness . . . 434
B.3. Disjunction and Proof Format . . . 435
B.4. Implication . . . 437
B.5. Proving Implications . . . 438
B.6. Free Variables and Substitution . . . 439
B.7. Universal Quantification . . . 441
B.8. Existential Quantification . . . 442
**C. Answers to Select Exercises** 445
**References** 459
**Index** 467