

## Table of Contents

1. Introduction .....	1
1.1 The Purpose of Abstraction	1
1.2 Principles	4
1.3 Languages	4
1.4 Learning Goals	5
1.5 Structure of the Chapters	6
1.6 How the Problems are Rated	8
1.7 Further Reading	9
1.8 Acknowledgment	10
2. Exploring Abstractions: Information Structures.....	11
2.1 Introduction	11
2.2 The Meaning of “Structure”	16
2.3 Homogeneous List Structures	16
2.4 Decomposing a List	17
2.5 List Manipulation	19
2.6 Lists of Lists	23
2.7 Binary Relations, Graphs, and Trees	25
2.8 Matrices	40
2.9 Undirected graphs	43
2.10 Indexing Lists vs. Arrays	44
2.11 Structure Sharing	45
2.12 Abstraction, Representation, and Presentation	48
2.13 Abstract Information Structures	52
2.14 Conclusion	54
2.15 Chapter Review	54
3. High-Level Functional Programming .....	57
3.1 Introduction	57
3.2 Nomenclature	58
3.3 Using High-Level Functions	61
3.4 Mapping, Functions as Arguments	66
3.5 Anonymous Functions	67
3.6 Functions as Results	69
3.7 Composing Functions	71
3.8 The Pipelining Principle	72
3.9 Functions of Functions in Calculus (Advanced)	73
3.10 Type Structure	75
3.11 Transposing Lists of Lists	78
3.12 Predicates	79
3.13 Generalized Sorting	82
3.14 Reducing Lists	82
3.15 Sequences as Functions	85
3.16 Solving Complex Problems using Functions	86
3.17 Conclusion	95
3.18 Chapter Review	95
4. Low-Level Functional Programming.....	97

4.1 Introduction	97
4.2 List-matching	97
4.3 Single-List Inductive Definitions	100
4.3 Rules Defining Functions	104
4.4 Lists vs. Numbers	106
4.5 Guarded Rules	109
4.6 Conditional Expressions	112
4.7 Equations	112
4.8 Equational Guards	112
4.9 More Recursion Examples	115
4.10 Accumulator-Argument Definitions	130
4.11 Interface vs. auxiliary functions	131
4.12 Tail Recursion	132
4.13 Using Lists to Implement Sets	135
4.14 Searching Trees	138
4.15 Searching Graphs	143
4.16 Argument Evaluation Disciplines	147
4.17 Infinite Lists (Advanced)	151
4.18 Perspective: Drawbacks of Functional Programming	162
4.19 Chapter Review	163
4.20 Further Reading	163
5. Implementing Information Structures .....	165
5.1 Introduction	165
5.2 Linked Lists	165
5.3 Open Lists	168
5.4 Closed Lists	173
5.5 Hashing	180
5.6 Principle of Virtual Contiguity	182
5.7 Chapter Review	184
5.8 Further Reading	184
6. States and Transitions .....	185
6.1 Introduction	185
6.2 Transition Relations	187
6.3 Transition Rule Notation	191
6.4 Rules for Assignment-Based Programs	201
6.5 Turing Machines – Simple Rewriting Systems	211
6.6 Turing's Thesis	215
6.7 Expressing Turing Machines as Rewrite Rules	220
6.8 Chapter Review	224
6.9 Further Reading	225
7. Object-Oriented Programming .....	227
7.1 Introduction	227
7.2 Classes	228
7.3 Attributes	229
7.4 Object Allocation	230
7.5 Static vs. Instance Variables	231

7.6 Example – A Stack Class	232
7.7 Stack Implementation	234
7.8 Improved Stack Implementation	236
7.9 Classes of Data Containers	239
7.10 Implementing a Queue as a Circular Array	240
7.11 Code Normalization and Class Hierarchies	243
7.12 Inheritance	245
7.12 Inheritance vs. Composition	250
7.13 The <i>is-a</i> Concept and Sub-Classing	252
7.14 Using Inheritance for Interface Abstraction	253
7.15 Abstract Classes	256
7.16 The <i>Object</i> Class	259
7.17 Wrapper Classes	260
7.18 Copying Objects	261
7.19 Equality for Objects	261
7.20 Principle of Interning	262
7.21 Linked-Lists in Object-Oriented Form	263
7.22 Enumeration Interfaces	267
7.23 Higher-Order Functions as Objects	270
7.24 Conclusion	272
7.25 Chapter Review	273
7.26 Further Reading	273
8. Induction, Grammars, and Parsing.....	275
8.1 Introduction	275
8.2 Using Rules to Define Sets	275
8.3 Languages	283
8.4 Structure of Grammars	284
8.5 The Relationship of Grammars to Meaning	290
8.6 Syntax Diagrams	300
8.7 Grouping and Precedence	303
8.8 Programs for Parsing	304
8.9 More on Expression Syntax vs. Semantics	312
8.10 Syntax-Directed Compilation (Advanced)	314
8.11 Varieties of Syntax	315
8.12 Chapter Review	324
8.13. Further Reading	325
9. Proposition Logic.....	327
9.1 Introduction	327
9.2 Encodings of Finite Sets	328
9.3 Encodings in Computer Synthesis	335
9.4 Propositions	339
10. Predicate Logic .....	379
10.1 Introduction	379
10.2 A Database Application	381
10.3 Backtracking Principle	388
10.4 Using Logic to Specify and Reason about Program Properties	397

10.5 Use of Assertions as a Programming Device	414
10.6 Program Proving vs. Program Testing	414
10.7 Using Assertional Techniques for Reasoning	415
10.8 Chapter Review	419
10.9 Further Reading	420
11. Complexity .....	421
11.1 Introduction	421
11.2 Resources	421
11.3 The Step-Counting Principle	422
11.4 Profiling	426
11.5 Suppressing Multiplicative Constants	427
11.6 Counting Dominant Operations	427
11.7 Growth-Rate	427
11.8 Upper Bounds	431
11.9 Asymptotic Growth-Rate	431
11.10 The "O" Notation	432
11.11 O Notation Rules	435
11.12 Analyzing Growth of Exotic Functions	438
11.13 Derivative Rule	440
11.14 Order-of-Magnitude Comparisons	441
11.15 Doubling Comparisons	442
11.16 Estimating Complexity Empirically	443
11.17 Case Studies Using Sorting	449
11.18 Complexity of Set Operations and Mappings	463
11.19 Chapter Review	469
11.20 Further Reading	469
12. Finite-State Machines.....	471
12.1 Introduction	471
12.2 Finite-State Grammars and Non-Deterministic Machines	496
12.3 Meaning of Regular Expressions	501
12.4 Synthesizing Finite-State Machines from Logical Elements	514
12.6 Inside Flip-Flops	528
12.7 The Progression to Computers	532
12.8 Chapter Review	544
12.9 Further Reading	545
13. Stored-Program Computers .....	547
13.1 Introduction	547
13.2 Programmer's Abstraction for a Stored-Program Computer	547
13.3 Examples of Machine-Level Programs	552
13.4 Memory-mapped I/O	561
13.5 Finite-State Control Unit of a Computer Processor	564
13.6 Forms of Addressing in Other Machines	571
13.7 Processor-Memory Communication	574
13.8 Interrupts	577
13.9 Direct Memory Access I/O	579
13.10 Pipelining within the Processor	580

13.11 Virtual Memory	580
13.12 Processor's Relation to the Operating System	581
13.13 Chapter Review	582
13.14 Further Reading	583
14. Parallel Computing.....	585
14.1 Introduction	585
14.2 Independent Parallelism	585
14.3 Scheduling	587
14.4 Stream Parallelism	588
14.5 Expression Evaluation	590
14.6 Data-Parallel Machines	592
14.7 Control-Parallel Machines	597
14.8 Distributed Memory Multiprocessors	603
14.9 Speedup and Efficiency	606
14.9 Chapter Review	606
14.10 Further Reading	607
15. Limitations to Computing.....	609
15.1 Introduction	609
15.2 Algorithm lower bounds	609
15.3 Limitations on Specific Classes of Machines	610
15.4 The Halting Problem	611
15.5 NP-Complete Problems	614
15.6 Amdahl's Law	615
15.7 The Glitch Phenomenon	616
15.8 Chapter Review	619
15.9 Further Reading	619
Index.....	621