Contents

Preface v

1 Aspects of Programming Languages 1

1.1 Why programming languages? 2
1.2 Some history 4
1.3 Paradigms and the structure of this book 6
1.3.1 Programming paradigms 8
1.3.2 Programming models 9
1.3.3 Other language classifications 10
1.4 The structure of programs 11
1.4.1 Lexical structure 12
1.4.2 Syntactic structure 14
1.4.3 Context and semantics 16
1.5 Programming languages as communication media 16
1.6 Managing and reducing complexity 18
1.6.1 Problem decomposition 18
1.6.2 Abstraction 19
1.6.3 Contextual checking 21
1.7 Program processing 23
1.7.1 Interpretation versus compilation 23
1.7.2 Macro preprocessing 26
1.7.3 Debugging tools 27
1.7.4 Programming environments 28
Summary 29
Bibliographical notes 29

Exercises 30

2 Imperative Languages 32

2.1 Principles 33

2.2 Data 34
2.2.1 Data declarations 34
2.2.2 Types and type constructors 38
2.2.3 Orthogonality of data types and declarations 50
2.2.4 Restricted types 52
2.2.5 Type equivalence 52
2.2.6 Coercions, contexts, casts and conversions 54

2.3 State 55
2.3.1 Assignments 56
2.3.2 Expressions 58
2.3.3 External state 62

2.4 Flow of control 64
2.4.1 Sequencing 64
2.4.2 Selection 64
2.4.3 Routine invocation 67
2.4.4 Repetition 67
2.4.5 Run-time error handling 71
2.4.6 Orthogonality of statements 76

2.5 Program composition 77
2.5.1 Blocks 78
2.5.2 Subprograms 84
2.5.3 Modules and packages 92
2.5.4 Generics 100
2.5.5 Programs 102
2.5.6 Orthogonality of program composition 105

2.6 Examples of imperative languages 105
2.6.1 Icon 105
2.6.2 PostScript 107

Summary 109

Bibliographical notes 110

Exercises 110

3 Object-oriented Languages 113

3.1 Principles 113
3.1.1 The object-oriented design methodology 113
3.1.2 The principles of object-oriented languages 115

3.2 Classes 116
3.3 Inheritance 118
3.4 Inheritance and class hierarchies 121
3.5 Inheritance and types 123
3.6 Inheritance and polymorphism 124
3.7 Dynamic binding 125
3.8 Reference semantics 128
3.9 When to use inheritance? 131
3.10 Discussion 132
3.11 Example languages 133
3.11.1 C++ 134
3.11.2 Eiffel 135
3.11.3 Smalltalk-80 136
Summary 139
Bibliographical notes 140
Exercises 140

4 Functional Languages 143
4.1 Principles 144
4.2 Functions 146
4.3 Lists 148
4.4 Types and polymorphism 150
4.5 Higher-order functions 151
4.6 Currying 153
4.7 Lazy evaluation 153
4.8 Equations and pattern matching 156
4.9 Example programs 158
4.10 Example language 161
4.10.1 Lisp 161
Summary 161
Bibliographical notes 162
Exercises 163

5 Logic Languages 165
5.1 Principles 165
5.2 Horn clauses 166

5.3 Executing Horn clauses 168
5.4 Logical variables 170
5.5 Relations 172
5.6 Data structures 173
5.7 Controlling the search order 176
5.8 Example programs 178
5.9 Example language 183
5.9.1 Prolog 183
Summary 184
Bibliographical notes 185
Exercises 185

6 Parallel and Distributed Languages 187
6.1 Principles 187
6.2 Parallelism 190
6.2.1 Co-routines 191
6.2.2 Parallel statements 191
6.2.3 Processes 193
6.3 Communication and synchronization 194
6.3.1 Shared variables 194
6.3.2 Message passing 202
6.3.3 In between message passing and shared variables 211
6.4 Languages based on other paradigms 214
6.5 Example languages 215
6.5.1 Mesa 216
6.5.2 Synchronizing Resources (SR) 216
6.5.3 Ada 217
Summary 219
Bibliographical notes 220
Exercises 221

7 Other Paradigms 223
7.1 Additional general-purpose paradigms 225
7.1.1 Constraint programming 225
7.1.2 Access-oriented programming 228
7.1.3 Single-datastructure languages 229
7.2 Additional special-purpose paradigms 232

7.2.1 Dataflow programming 232
7.2.2 Little languages 238
7.2.3 Database languages 240
7.2.4 Real-time languages 241

Summary 243

Bibliographical notes 244

Exercises 244

Appendix A: References to Languages 246

Appendix B: Answers to Exercises 249

References 258

Index 264