

About the Author	p. ii
Preface	p. xvii
Introduction	p. 1
The Art of Language Design	p. 3
The Programming Language Spectrum	p. 5
Why Study Programming Languages?	p. 7
Compilation and Interpretation	p. 9
Programming Environments	p. 14
An Overview of Compilation	p. 15
Lexical and Syntax Analysis	p. 16
Semantic Analysis and Intermediate Code Generation	p. 18
Target Code Generation	p. 22
Code Improvement	p. 24
Summary and Concluding Remarks	p. 24
Review Questions	p. 25
Exercises	p. 26
Bibliographic Notes	p. 28
Programming Language Syntax	p. 31
Specifying Syntax: Regular Expressions and Context-Free Grammars	p. 32
Tokens and Regular Expressions	p. 33
Context-Free Grammars	p. 34
Derivations and Parse Trees	p. 36
Recognizing Syntax: Scanners and Parsers	p. 39
Scanning	p. 40
Top-Down and Bottom-Up Parsing	p. 48
Recursive Descent	p. 51
Syntax Errors	p. 57
Table-Driven Top-Down Parsing	p. 62
Bottom-Up Parsing	p. 75
Theoretical Foundations	p. 87
Finite Automata	p. 88
Push-Down Automata	p. 92
Grammar and Language Classes	p. 93
Summary and Concluding Remarks	p. 94
Review Questions	p. 97
Exercises	p. 98
Bibliographic Notes	p. 102
Names, Scopes, and Bindings	p. 105
The Notion of Binding Time	p. 106
Object Lifetime and Storage Management	p. 108
Stack-Based Allocation	p. 111

Heap-Based Allocation	p. 113
Garbage Collection	p. 114
Scope Rules	p. 115
Static Scope	p. 116
Dynamic Scope	p. 129
Symbol Tables	p. 132
Association Lists and Central Reference Tables	p. 137
The Binding of Referencing Environments	p. 139
Subroutine Closures	p. 141
First- and Second-Class Subroutines	p. 143
Overloading and Related Concepts	p. 144
Naming-Related Pitfalls in Language Design	p. 149
Scope Rules	p. 149
Separate Compilation	p. 151
Summary and Concluding Remarks	p. 155
Review Questions	p. 157
Exercises	p. 158
Bibliographic Notes	p. 162
Semantic Analysis	p. 165
The Role of the Semantic Analyzer	p. 166
Attribute Grammars	p. 168
Attribute Flow	p. 170
Action Routines	p. 179
Space Management for Attributes	p. 180
Bottom-Up Evaluation	p. 181
Top-Down Evaluation	p. 186
Annotating a Syntax Tree	p. 191
Summary and Concluding Remarks	p. 197
Review Questions	p. 198
Exercises	p. 199
Bibliographic Notes	p. 202
Assembly-Level Computer Architecture	p. 203
Workstation Macro-Architecture	p. 204
The Memory Hierarchy	p. 207
Data Representation	p. 209
Integer Arithmetic	p. 211
Floating-Point Arithmetic	p. 212
Instruction Set Architecture	p. 214
Addressing Modes	p. 215
Conditional Branches	p. 217
The Evolution of Processor Architecture	p. 218

Two Example Architectures: The 680x0 and MIPS	p. 220
Pseudoassembler Notation	p. 225
Compiling for Modern Processors	p. 227
Keeping the Pipeline Full	p. 227
Register Allocation	p. 234
Summary and Concluding Remarks	p. 242
Review Questions	p. 243
Exercises	p. 244
Bibliographic Notes	p. 247
Control Flow	p. 249
Expression Evaluation	p. 250
Precedence and Associativity	p. 251
Assignments	p. 254
Ordering Within Expressions	p. 262
Short-Circuit Evaluation	p. 265
Structured and Unstructured Flow	p. 267
Sequencing	p. 270
Selection	p. 271
Short-Circuited Conditions	p. 272
Case/Switch Statements	p. 275
Iteration	p. 280
Enumeration-Controlled Loops	p. 280
Combination Loops	p. 286
Iterators	p. 287
Logically Controlled Loops	p. 294
Recursion	p. 297
Iteration and Recursion	p. 297
Applicative- and Normal-Order Evaluation	p. 301
Nondeterminacy	p. 303
Summary and Concluding Remarks	p. 308
Review Questions	p. 310
Exercises	p. 311
Bibliographic Notes	p. 316
Data Types	p. 319
Type Systems	p. 320
The Definition of Types	p. 322
The Classification of Types	p. 323
Type Checking	p. 330
Type Equivalence	p. 330
Type Conversion and Casts	p. 334
Type Compatibility and Coercion	p. 337

Type Inference	p. 341
The ML Type System	p. 344
Records (Structures) and Variants (Unions)	p. 351
Syntax and Operations	p. 351
Memory Layout and Its Impact	p. 353
With Statements	p. 355
Variant Records	p. 358
Arrays	p. 365
Syntax and Operations	p. 365
Dimensions, Bounds, and Allocation	p. 369
Memory Layout	p. 373
Strings	p. 379
Sets	p. 381
Pointers and Recursive Types	p. 382
Syntax and Operations	p. 383
Dangling References	p. 391
Garbage Collection	p. 395
Lists	p. 401
Files and Input/Output	p. 403
Interactive I/O	p. 404
File-Based I/O	p. 405
Text I/O	p. 407
Equality Testing and Assignment	p. 414
Summary and Concluding Remarks	p. 416
Review Questions	p. 418
Exercises	p. 420
Bibliographic Notes	p. 425
Subroutines and Control Abstraction	p. 427
Review of Stack Layout	p. 428
Calling Sequences	p. 431
Case Study: C on the MIPS	p. 434
Case Study: Pascal on the 680x0	p. 437
In-Line Expansion	p. 441
Parameter Passing	p. 442
Parameter Modes	p. 443
Special-Purpose Parameters	p. 453
Function Returns	p. 457
Generic Subroutines and Modules	p. 459
Exception Handling	p. 464
Definition of Exceptions	p. 466
Exception Propagation	p. 468

Example: Phrase-Level Recovery in a Recursive Descent Parser	p. 470
Implementation of Exceptions	p. 471
Coroutines	p. 474
Stack Allocation	p. 476
Transfer	p. 478
Iterators	p. 479
Example: Discrete Event Simulation	p. 480
Summary and Concluding Remarks	p. 484
Review Questions	p. 485
Exercises	p. 486
Bibliographic Notes	p. 489
Building a Runnable Program	p. 491
Back-End Compiler Structure	p. 491
An Example	p. 492
Phases and Passes	p. 496
Intermediate Forms	p. 496
Diana	p. 498
GNU RTL	p. 499
Code Generation	p. 503
An Attribute Grammar Example	p. 504
Register Allocation	p. 504
Address Space Organization	p. 507
Assembly	p. 510
Emitting Instructions	p. 511
Assigning Addresses to Names	p. 514
Linking	p. 515
Relocation and Name Resolution	p. 515
Type Checking	p. 516
Dynamic Linking	p. 518
Position-Independent Code	p. 519
Fully Dynamic (Lazy) Linking	p. 521
Summary and Concluding Remarks	p. 522
Review Questions	p. 523
Exercises	p. 524
Bibliographic Notes	p. 527
Data Abstraction and Object Orientation	p. 529
Object-Oriented Programming	p. 530
Encapsulation and Inheritance	p. 539
Modules	p. 539
Classes	p. 542
Type Extensions	p. 544

Intitialization and Finalization	p. 546
Choosing a Constructor	p. 547
References and Values	p. 550
Execution Order	p. 551
Garbage Collection	p. 553
Dynamic Method Binding	p. 554
Virtual and Nonvirtual Methods	p. 555
Abstract Classes	p. 557
Member Lookup	p. 557
Related Concepts	p. 561
Multiple Inheritance	p. 564
Semantic Ambiguities	p. 568
Replicated Inheritance	p. 570
Shared Inheritance	p. 572
Mix-In Inheritance	p. 573
Object-Oriented Programming Revisited	p. 574
The Object Model of Smalltalk	p. 577
Summary and Concluding Remarks	p. 580
Review Questions	p. 582
Exercises	p. 583
Bibliographic Notes	p. 586
Nonimperative Programming Models: Functional and Logic Languages	p. 589
Historical Origins	p. 590
Functional Programming	p. 592
A Review/Overview of Scheme	p. 594
Evaluation Order Revisited	p. 604
Higher-Order Functions	p. 609
Theoretical Foundations	p. 612
Functional Programming in Perspective	p. 622
Logic Programming	p. 624
Prolog	p. 625
Theoretical Foundations	p. 641
Logic Programming in Perspective	p. 646
Summary and Concluding Remarks	p. 648
Review Questions	p. 650
Exercises	p. 651
Bibliographic Notes	p. 657
Concurrency	p. 659
Background and Motivation	p. 660
A Little History	p. 660
The Case for Multithreaded Programs	p. 663

Multiprocessor Architecture	p. 667
Concurrent Programming Fundamentals	p. 670
Communication and Synchronization	p. 671
Languages and Libraries	p. 672
Thread Creation Syntax	p. 673
Implementation of Threads	p. 682
Shared Memory	p. 687
Busy-Wait Synchronization	p. 688
Scheduler Implementation	p. 692
Scheduler-Based Synchronization	p. 694
Implicit Synchronization	p. 703
Message Passing	p. 706
Naming Communication Partners	p. 706
Sending	p. 710
Receiving	p. 714
Remote Procedure Call	p. 719
Summary and Concluding Remarks	p. 722
Review Questions	p. 724
Exercises	p. 725
Bibliographic Notes	p. 730
Code Improvement	p. 733
Phases of Code Improvement	p. 735
Peephole Optimization	p. 737
Redundancy Elimination in Basic Blocks	p. 740
Global Redundancy and Data Flow Analysis	p. 745
Static Single Assignment Form and Global Value Numbering	p. 746
Global Common Subexpression Elimination	p. 750
Loop Improvement I	p. 755
Loop Invariants	p. 755
Induction Variables	p. 756
Instruction Scheduling	p. 759
Loop Improvement II	p. 763
Loop Unrolling and Software Pipelining	p. 763
Loop Reordering	p. 767
Register Allocation	p. 775
Summary and Concluding Remarks	p. 778
Review Questions	p. 780
Exercises	p. 781
Bibliographic Notes	p. 785
Programming Languages Mentioned	p. 787
Language Design and Language Implementation	p. 795

Bibliography

p. 801

Index

p. 827

Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.