Contents

Introduction **i**

1 Foundations **1**

1.1 Prerequisites 1
1.2 Foundations of Programming 1
1.3 Simple Data Types 2
1.3.1 Integers in Binary 2
1.3.2 Floating-point Numbers 4
1.3.3 Characters and ASCII 4
1.3.4 Strings 5
1.3.5 Representations and Types 5
1.3.6 Type Systems and Type Checking 6
1.3.7 Summary 7
1.4 Variables 8
1.4.1 Assignment 8
1.4.2 Mutable and immutable variables 10
1.4.3 Summary 13
1.5 Control structures 14
1.5.1 Conventional control structures 14
1.5.2 Control structures in computers 17
1.5.3 Summary 18
1.6 Functions 20
1.6.1 Calling functions 20
1.6.2 Summary 23
1.7 Practice 24

2 Understanding VMs **29**

2.1 Prerequisites 29
2.2 Simple VMs 29
2.2.1 The stack 30

2.2.2 Interpreter technique 31
2.2.3 VM1 implementation 32
2.2.4 REGVM implementation 33
2.2.5 Portability 36
2.2.6 Summary 38
2.3 Stack-based VM 39
2.3.1 Comparisons 41
2.3.2 Iterations 42
2.3.3 Error handling 43
2.3.4 Summary 46
2.4 Memory and functions 46
2.4.1 Frame pointer 51
2.4.2 Local storage 53
2.4.3 Memory management 54
2.4.4 Frame stack 56
2.4.5 Summary 61
2.5 Practice 63

3 Development Environment 67
3.1 Prerequisites 67
3.2 Basic tools 67
3.3 Debugging 68
3.3.1 Process 70
3.3.2 Root Cause Analysis 71
3.3.3 Tools 74
3.3.4 Summary 81
3.4 Optimisation 81
3.4.1 Memory 86
3.4.2 Time 88
3.4.3 Dynamic Programming 90
3.4.4 Confusion matrix 92
3.4.5 Summary 98
3.5 Tests and testing 98
3.5.1 Automated testing and continuous integration 103
3.5.2 Summary 108
3.6 Advanced Debugging Practices 108
3.6.1 Error Handling and Resilience 109
3.6.2 Production and Live System Debugging 115
3.6.3 Summary 127
3.7 The Impact of LLMs 127
3.7.1 Debugging 127

3.7.2 Optimisation 128
3.7.3 Testing 129
3.7.4 Broader Perspectives and Challenges 129
3.7.5 Summary 130
3.8 Practice 130

4 Building and Experimenting 133
4.1 Prerequisites 133
4.2 The computer as hardware 133
4.2.1 Hardware 136
4.2.2 The Pico 138
4.2.3 Summary 141
4.2.4 Practice 142
4.3 Basic Input/Output 142
4.3.1 The Pico pins 144
4.3.2 Light switching circuit 145
4.3.3 Programmable I/O 146
4.3.4 State machines 148
4.3.5 Circuit for traffic lights 151
4.3.6 Pedestrian crossing 153
4.3.7 Temperature measurements 155
4.3.8 Temperature indicator 156
4.3.9 Summary 162
4.3.10 Practice 164
4.4 Secondary memory 166
4.4.1 Save and load temperatures 169
4.4.2 A simple database on SD card 170
4.4.3 Summary 173
4.4.4 Practice 173
4.5 External Communication 174
4.5.1 Universal Asynchronous Receiver-Transmitter 175
4.5.2 Inter-Pico Communication 179
4.5.3 Client-Server Architecture 183
4.5.4 Wireless Communication 190
4.5.5 WiFi Communication Between Pico W Boards 191
4.5.6 Advanced WiFi Features 196
4.5.7 Features 199
4.5.8 Network Protocol Stack Integration 200
4.5.9 Socket Programming 200
4.5.10 Summary 205
4.5.11 Practice 206

4.6 Cryptography and Secure Communication . . . 208
4.6.1 Core Principles . . . 208
4.6.2 Symmetric Encryption . . . 209
4.6.3 Asymmetric Encryption . . . 210
4.6.4 Data Integrity and Hashing . . . 217
4.6.5 Hybrid Cryptographic Systems . . . 218
4.6.6 Digital Signatures . . . 221
4.6.7 Implementation Considerations . . . 221
4.6.8 Summary . . . 223
4.6.9 Practice . . . 223
4.7 Graphics . . . 225
4.7.1 Framebuffers and DMA . . . 226
4.7.2 Raster and Vector Approaches . . . 229
4.7.3 Display Interfaces . . . 232
4.7.4 Pimoroni Pico Display Pack 2.0 . . . 232
4.7.5 Colour Models and Bit Depth . . . 235
4.7.6 Rendering Strategies . . . 236
4.7.7 Simple Double Buffering Example . . . 237
4.7.8 Performance Considerations . . . 237
4.7.9 Text Rendering . . . 238
4.7.10 Special Techniques and Hardware . . . 239
4.7.11 Summary . . . 241
4.7.12 Practice . . . 242
4.8 Handling errors . . . 243
4.8.1 Error Types . . . 243
4.8.2 Approaches . . . 244
4.8.3 Summary . . . 248
4.8.4 Practice . . . 248
4.9 Timing and timers . . . 248
4.9.1 Timing and Timers in the Pico . . . 249
4.9.2 Interrupt handling . . . 251
4.9.3 Summary . . . 251
4.9.4 Practice . . . 251
4.10 Concurrency and multithreading . . . 252
4.10.1 Summary . . . 257
4.10.2 Practice . . . 257
4.11 Power management . . . 257
4.11.1 Summary . . . 262
4.11.2 Practice . . . 262
4.12 Practice . . . 262

5 The Compiler Pipeline 267
5.1 Prerequisites . . . 267
5.2 Compilers, Interpreters, Assembler, and VMs . . . 267
5.3 A Short History . . . 270
5.4 The Pipeline in Principle . . . 271
5.5 Syntax . . . 272
5.5.1 Tokenisation or Lexical Analysis . . . 273
5.5.2 Parsing . . . 273
5.5.3 Production Rules and Grammars . . . 274
5.5.4 Summary . . . 275
5.6 Semantic Analysis . . . 275
5.6.1 How Semantic Analysis Works . . . 276
5.6.2 Symbol Table . . . 277
5.6.3 Type System . . . 278
5.6.4 Using Abstract Syntax Trees . . . 279
5.6.5 Control Flow Graph . . . 284
5.6.6 Example: Control Flow Graph . . . 287
5.6.7 Summary . . . 291
5.7 Intermediate Representations . . . 292
5.7.1 Static Single Assignment . . . 294
5.7.2 Three Address Code . . . 296
5.7.3 Directed Acyclic Graphs . . . 297
5.7.4 Summary . . . 301
5.8 Code Generation Techniques . . . 302
5.8.1 Example: Tail-Recursion . . . 310
5.8.2 Summary . . . 315
5.9 Loader and Linker . . . 316
5.9.1 Linkers . . . 316
5.9.2 Loaders . . . 316
5.9.3 Real Examples . . . 317
5.9.4 Simple Simulation . . . 317
5.9.5 Summary . . . 322
5.10 Optimisation in Compilers . . . 322
5.10.1 Types of Compiler Optimisations . . . 323
5.10.2 The Future of Compiler Optimisation . . . 327
5.10.3 Summary . . . 328
5.11 Error Handling . . . 328
5.11.1 Warnings . . . 329
5.11.2 Errors . . . 329
5.11.3 Future of Error Handling . . . 332
5.11.4 Summary . . . 333

5.12 Practice 334

6 Philosophy and Methodology 343

6.1 Philosophy and style 343

6.1.1 Programming as craft 344

6.1.2 Building process 345

6.1.3 Blending philosophies 348

6.1.4 Methodologies 351

6.1.5 References 353

6.1.6 A personal note 355

6.1.7 Risks 357

6.1.8 Summary 358

6.2 Problems and methods 359

6.2.1 Example: Low customer satisfaction 359

6.2.2 Example: Thermostat 360

6.2.3 Generalisations 361

6.2.4 Methodology in the craft philosophy 365

6.2.5 Summary 367

6.3 Reflections on the history and future 368

6.3.1 AI example: LLMs 370

6.3.2 Summary 373

6.4 Method examples 373

6.4.1 Mock-ups and prototypes 374

6.4.2 Code review 378

6.4.3 Null hypothesis in programming 386

6.4.4 Summary 390

6.5 Practice 391

7 The Mechanics 395

7.1 Data: Foundations, Structures, and Algorithms 395

7.1.1 Elementary Data Types 396

7.1.2 Data Structures 396

7.1.3 Abstract Data Types 397

7.1.4 Algorithms 398

7.1.5 Summary 398

7.2 Types and Type Systems 399

7.2.1 Data Types 399

7.2.2 Type Systems 399

7.2.3 Type Errors 399

7.2.4 Static vs Dynamic Typing 400

7.2.5 Type Inference 400

7.2.6 Type Consistency and Soundness 401
7.2.7 Strong vs. Weak Typing 401
7.2.8 Beyond Data Safety 402
7.2.9 Summary 402
7.3 Programming Paradigms and Language Design 403
7.3.1 Imperative Languages 404
7.3.2 Functional Languages 406
7.3.3 Object-Oriented Languages 409
7.3.4 Concatenative Languages 416
7.3.5 Homoiconicity and Metaprogramming 418
7.3.6 Domain-Specific Languages 419
7.3.7 Example: Rust 425
7.3.8 Summary 427
7.4 Low-Level Constructs 428
7.4.1 Callback 429
7.4.2 State Machine 431
7.4.3 Checkpoint 432
7.4.4 Low-Level Programming Concepts 434
7.4.5 Summary 436
7.5 Architecture in Programming 436
7.5.1 History 437
7.5.2 Styles and Patterns 437
7.5.3 Change 439
7.5.4 Code Architecture vs. System Architecture 440
7.5.5 Practical Considerations in Modern Architectures 440
7.5.6 Summary 441
7.6 Design Patterns 441
7.6.1 Factory 443
7.6.2 Observer 445
7.6.3 Command 448
7.6.4 Strategy 450
7.6.5 Adapter 453
7.6.6 Composite 455
7.6.7 Decorator 457
7.6.8 Summary 459
7.7 Networking and Modern Server Architectures 460
7.7.1 Server Systems 461
7.7.2 A Protocol Suite: TCP/IP 463
7.7.3 Web Servers 472
7.7.4 Application-Level Protocols and Styles 483
7.7.5 System-Level Architectural Patterns 485

7.7.6 Concurrency and Advanced Server Functions 489
7.7.7 Summary 492
7.8 Practice 493

8 Perspectives and Frontiers 497
8.1 Systemic Concepts in Computing 497
8.1.1 Abstraction 497
8.1.2 Latency 498
8.1.3 Resilience 499
8.1.4 Summary 500
8.2 Computer Security 502
8.2.1 The Modern Threat Landscape 503
8.2.2 Fundamental Principles 504
8.2.3 The Security Ecosystem 505
8.2.4 Human Factors 506
8.2.5 The Role of Cryptography 506
8.2.6 Principles of Cryptography 507
8.2.7 Cryptographic Techniques 508
8.2.8 Challenges in Cryptographic Security 508
8.2.9 Summary 509
8.3 Exploring the Limits of Computation 510
8.3.1 Summary 516
8.4 The Turing Machine 517
8.4.1 The Halting Problem 518
8.4.2 Proof of Undecidability 518
8.4.3 Visualising the Proof 520
8.4.4 Implications 522
8.4.5 Summary 522
8.5 Information Theory 523
8.5.1 Quantifying Information 523
8.5.2 Entropy 524
8.5.3 The Dance of Information 525
8.5.4 The Limit of Communication 526
8.5.5 The Art of Data Compression 528
8.5.6 Divergence and Cross-Entropy 528
8.5.7 Summary 529
8.6 Artificial Intelligence and Machine Learning 530
8.7 Machine Learning 534
8.7.1 Three Samples 541
8.7.2 Summary 553
8.8 Deep Learning 553

8.8.1 Foundations 556
8.8.2 Training via Backpropagation 557
8.8.3 Convolutional Neural Networks (CNNs) 558
8.8.4 Recurrent Neural Networks (RNNs) and LSTMs 559
8.8.5 Transformers 560
8.8.6 Summary 561
8.9 Capabilities, Limitations, and Impacts of AI/ML 562
8.9.1 Limitations and Risks 563
8.9.2 Strengths and Achievements 570
8.9.3 Summary 571
8.10 AI: The Catalyst for a New Era in Programming 572
8.10.1 The Quality-Time Tradeoff 573
8.10.2 Rethinking Verification 573
8.10.3 Formal Approaches 574
8.10.4 Empirical Approaches 574
8.10.5 Summary 575
8.11 Hoare Logic 576
8.11.1 Inference Rules 577
8.11.2 Practical Examples 579
8.11.3 Advanced Topics 582
8.11.4 Proof Techniques: Loop Invariants & Failed Proofs . . . 583
8.11.5 Tools and Applications 584
8.11.6 Limitations and Extensions 584
8.11.7 Summary 585
8.12 Property-Based Testing 585
8.12.1 The Idea 586
8.12.2 Properties vs. Examples: A Paradigm Shift 588
8.12.3 Foundations 589
8.12.4 Implementation Architecture 593
8.12.5 Advanced Concepts and Techniques 598
8.12.6 Practical Applications 605
8.12.7 Performance Considerations 616
8.12.8 Debugging and Diagnosis 618
8.12.9 Best Practices and Guidelines 621
8.12.10 Model-Based and Metamorphic Testing 622
8.12.11 Future Directions 626
8.12.12 Summary 626
8.13 Type Systems and Formal Reasoning 627
8.13.1 Lambda Calculus: A Simplified Overview 628
8.13.2 Dependent Types 630
8.13.3 Basic Concepts 630

8.13.4 Type Theory Basics 631
8.13.5 A Simple Type Theory with Dependent Types 634
8.13.6 Practical Applications and Examples 637
8.13.7 Example: Natural Deduction 638
8.13.8 Summary 641
8.14 Practice 641

Glossary **645**

Index **667**