

Computing with Data

Guy Lebanon • Mohamed El-Geish

Computing with Data

An Introduction to the Data Industry

www.computingwithdata.com

 Springer

Guy Lebanon
Amazon
Menlo Park
CA, USA

Mohamed El-Geish
Voicera
Santa Clara
CA, USA

ISBN 978-3-319-98148-2 ISBN 978-3-319-98149-9 (eBook)
<https://doi.org/10.1007/978-3-319-98149-9>

Library of Congress Control Number: 2018954275

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Anat Lebanon

Guy

*To my family and friends who put up with me
while writing this (and elsewhere).*

Mohamed

Contents

1	Introduction: How to Use This Book?	1
	References.....	5
2	Essential Knowledge: Hardware	7
2.1	RAM and ROM.....	7
2.2	The Disk.....	8
2.3	The Central Processing Unit.....	9
2.4	The Clock.....	10
	2.4.1 Logical and Physical Clocks.....	11
	2.4.2 Clock Drift.....	12
2.5	The Graphics Processing Unit.....	13
2.6	Binary Representations.....	14
	2.6.1 Binary Representation of Integers.....	14
	2.6.2 Binary Representation of Real Numbers.....	16
	2.6.3 Encoding Strings as Bits.....	17
	2.6.4 Rounding, Overflow, and Underflow.....	17
2.7	Assembly Language.....	21
	2.7.1 Memory Addresses.....	21
	2.7.2 Instruction Set.....	22
2.8	Interrupts.....	27
2.9	The Memory Hierarchy.....	28
	2.9.1 Cache Structure.....	30
	2.9.2 Direct Mapping and Associativity.....	32
	2.9.3 Cache Miss.....	33
	2.9.4 Cache Hierarchy.....	33
2.10	Multicores and Multiprocessors Computers.....	34
2.11	Notes.....	35
	References.....	36

3	Essential Knowledge: Operating Systems	37
3.1	Windows, Linux, and macOS	38
3.2	Command-Line Interfaces	39
3.2.1	The Linux Terminal and Bash	39
3.2.2	Command Prompt in Windows.....	46
3.2.3	PowerShell	54
3.3	The Kernel, Traps, and System Calls	65
3.4	Process Management	67
3.4.1	Processes in Linux	67
3.4.2	Processes in Windows	72
3.5	Memory Management and Virtual Memory	73
3.6	The File System.....	74
3.6.1	Files in Linux	75
3.6.2	Files in Windows.....	85
3.7	Users and Permissions	89
3.7.1	Users and Permissions in Linux.....	89
3.7.2	Users and Permissions in Windows	91
3.8	Input and Output	92
3.8.1	Redirecting Input and Output in Linux	93
3.8.2	Redirecting Input and Output in Windows	94
3.9	Networking.....	95
3.9.1	Working on Remote Linux Computers	95
3.9.2	Working on Remote Windows Computers.....	97
3.10	Notes.....	98
	References.....	98
4	Learning C++	99
4.1	Compilation	100
4.2	Types, Variables, and Scope	102
4.2.1	Types	103
4.2.2	Variables.....	103
4.2.3	Scope	105
4.3	Operators and Casting	106
4.3.1	Operators	106
4.3.2	Type Conversions	108
4.4	References and Pointers	109
4.4.1	References.....	109
4.4.2	Pointers	110
4.5	Arrays	111
4.5.1	One-Dimensional Arrays	111
4.5.2	Multidimensional Arrays	113
4.6	Preprocessor and Namespaces	113
4.7	Strings, Input, and Output	116

- 4.8 Control Flow 118
 - 4.8.1 If-Else Clauses 118
 - 4.8.2 While-Loops 120
 - 4.8.3 For-Loops 121
- 4.9 Functions 124
 - 4.9.1 Return Value 124
 - 4.9.2 Function Parameters 125
 - 4.9.3 Function Definition and Function Declaration 125
 - 4.9.4 Scope of Function Variables 127
 - 4.9.5 Pointer and Reference Parameters 127
 - 4.9.6 Recursion 129
 - 4.9.7 Passing Arguments to Main 132
 - 4.9.8 Overloading Functions 133
- 4.10 Object Oriented Programming 133
 - 4.10.1 Structs 134
 - 4.10.2 Classes 140
 - 4.10.3 Encapsulation 147
 - 4.10.4 Inheritance 148
 - 4.10.5 Polymorphism 150
 - 4.10.6 Static Variables and Functions 153
- 4.11 Dynamic Memory and Smart Pointers 154
 - 4.11.1 Dynamic Memory Allocation 154
 - 4.11.2 Smart Pointers 156
- 4.12 Templates 157
 - 4.12.1 Template Functions 158
 - 4.12.2 Template Classes 160
- 4.13 The Standard Template Library 162
 - 4.13.1 Sequence Containers 162
 - 4.13.2 Associative Containers 164
 - 4.13.3 Unordered Containers 166
- 4.14 Notes 167
- References 168
- 5 Learning Java 169**
 - 5.1 Compilation 170
 - 5.2 Types, Variables, and Scope 172
 - 5.3 Operators and Casting 173
 - 5.4 Primitive and Non-Primitive Types 173
 - 5.5 Arrays 175
 - 5.5.1 One-Dimensional Arrays 175
 - 5.5.2 Multidimensional Arrays 176
 - 5.6 Packages and the Import Statement 177
 - 5.7 Strings, Input, and Output 178
 - 5.8 Control Flow 179
 - 5.9 Functions 179

- 5.10 Object Oriented Programming 180
 - 5.10.1 Classes 180
 - 5.10.2 Inheritance 183
 - 5.10.3 Abstract Classes 184
 - 5.10.4 Access Modifiers 185
- 5.11 The Object Class 185
- 5.12 Interfaces 186
- 5.13 Generics 186
- 5.14 Collections 188
- 5.15 Notes 190
- References 190
- 6 Learning Python and a Few More Things 191**
 - 6.1 Objects 192
 - 6.2 Scalar Data Types and Operators 194
 - 6.2.1 Strings 196
 - 6.2.2 Duck Typing 198
 - 6.3 Compound Data Types 199
 - 6.3.1 Tuples 199
 - 6.3.2 Lists 200
 - 6.3.3 Ranges 201
 - 6.3.4 Slicing 202
 - 6.3.5 Sets 203
 - 6.3.6 Dictionaries 204
 - 6.4 Comprehensions 209
 - 6.4.1 List Comprehensions 209
 - 6.4.2 Set Comprehensions 210
 - 6.4.3 Dictionary Comprehensions 211
 - 6.4.4 Nested Comprehensions 211
 - 6.5 Control Flow 212
 - 6.5.1 If-Else 212
 - 6.5.2 For-Loops 212
 - 6.5.3 Else as a Completion Clause 213
 - 6.5.4 The Empty Statement 214
 - 6.6 Functions 215
 - 6.6.1 Anonymous Functions 221
 - 6.7 Classes 223
 - 6.7.1 Inheritance 225
 - 6.7.2 The Empty Class 226
 - 6.8 IPython 227
 - 6.8.1 Debugging 228
 - 6.8.2 Profiling 228
 - 6.9 NumPy, SciPy, Pandas, and scikit-learn 229
 - 6.9.1 Narray Objects 230
 - 6.9.2 Linear Algebra and Random Number Generation 234

6.9.3	Sparse Matrices in Python	237
6.9.4	Dataframes	239
6.9.5	scikit-learn	242
6.10	Reading and Writing to Files	247
6.10.1	Reading and Writing Data in Text Format	247
6.10.2	Reading and Writing Ndarrays in Binary Format	248
6.10.3	Reading and Writing Ndarrays in Text Format	249
6.10.4	Reading and Writing Dataframes	250
6.11	Material Differences Between Python 3.x and 2.x	251
6.11.1	Unicode Support	251
6.11.2	Print	251
6.11.3	Division	252
6.12	Notes.....	253
	References.....	253
7	Learning R	255
7.1	R, Matlab, and Python	255
7.2	Getting Started	256
7.3	Scalar Data Types	261
7.4	Vectors, Arrays, Lists, and Dataframes	262
7.5	If-Else, Loops, and Functions	268
7.6	Interfacing with C++ Code	271
7.7	Customization	275
7.8	Notes.....	276
	Reference.....	276
8	Visualizing Data in R and Python	277
8.1	Graphing Data in R	277
8.2	Datasets.....	278
8.3	Graphics and ggplot2 Packages	279
8.4	Strip Plots	280
8.5	Histograms	281
8.6	Line Plots.....	284
8.7	Smoothed Histograms	287
8.8	Scatter Plots	295
8.9	Contour Plots.....	308
8.10	Quantiles and Box Plots	310
8.11	qq-Plots.....	312
8.12	Devices	315
8.13	Data Preparation	317
8.14	Python’s Matplotlib Module.....	318
8.14.1	Figures.....	319
8.14.2	Scatter-Plots, Line-Plots, and Histograms	320
8.14.3	Contour Plots and Surface Plots.....	321

- 8.15 Notes..... 324
- References..... 324
- 9 Processing Data in R and Python 325**
 - 9.1 Missing Data 325
 - 9.1.1 Missing Data in R..... 327
 - 9.1.2 Missing Data in Python..... 329
 - 9.2 Outliers 331
 - 9.3 Data Transformations 334
 - 9.3.1 Skewness and Power Transformation 334
 - 9.3.2 Binning 341
 - 9.3.3 Indicator Variables 343
 - 9.4 Data Manipulation 344
 - 9.4.1 Random Sampling, Partitioning, and Shuffling 344
 - 9.4.2 Concatenations and Joins 346
 - 9.4.3 Tall Data and Wide Data..... 349
 - 9.4.4 Reshaping Data 350
 - 9.4.5 The Split-Apply-Combine Framework 354
 - 9.5 Notes..... 360
 - References..... 361
- 10 Essential Knowledge: Parallel Programming 363**
 - 10.1 Choosing a Programming Language 364
 - 10.2 Processes, Threads, and Fibers 365
 - 10.3 Thread Safety 365
 - 10.4 Volatility 368
 - 10.5 Synchronization 369
 - 10.5.1 Ineffectual Synchronization 370
 - 10.5.2 Synchronization vs. Volatility 373
 - 10.6 Starvation 374
 - 10.7 Deadlocks 376
 - 10.8 The Producer-Consumer Problem..... 379
 - 10.9 Reader-Writer Locks..... 383
 - 10.10 Reentrant Locks 388
 - 10.10.1 Reentry of Intrinsic Locks 392
 - 10.11 Higher-Level Concurrency Constructs and Frameworks..... 392
 - 10.11.1 Executors 393
 - 10.11.2 ParSeq 398
 - 10.11.3 Inter-Process Communication and Synchronization ... 404
 - 10.12 Non-Blocking Parallel Computing 410
 - 10.13 Beyond the CPU 411
 - 10.14 Notes..... 412
 - 10.14.1 Python 412
 - 10.14.2 Further Readings 413
 - References..... 413

- 11 Essential Knowledge: Testing** 415
 - 11.1 Black-Box Testing 416
 - 11.2 White-Box Testing 417
 - 11.3 Gray-Box Testing 418
 - 11.4 Levels of Testing 419
 - 11.5 Unit Testing 420
 - 11.5.1 Planning and Equivalence Class Partitioning 422
 - 11.5.2 Code Coverage 422
 - 11.5.3 Coding for Testability 423
 - 11.5.4 Mocking 423
 - 11.5.5 Test Hooks 426
 - 11.5.6 Test Case Anatomy 431
 - 11.5.7 Smoke Testing 432
 - 11.5.8 Happy-Path Testing 433
 - 11.5.9 Data-Driven Testing 433
 - 11.5.10 Fuzzing 434
 - 11.6 Integration Testing 434
 - 11.7 System Testing 435
 - 11.7.1 Performance Testing 435
 - 11.7.2 Load Testing 436
 - 11.7.3 Stress Testing 436
 - 11.8 Acceptance Testing 436
 - 11.9 Real-User Testing 437
 - 11.9.1 Canary Deployments 437
 - 11.10 Notes 439
 - References 439

- 12 A Few More Things About Programming** 441
 - 12.1 Notebooks 441
 - 12.2 Version Control 441
 - 12.2.1 Git 443
 - 12.2.2 GitHub 452
 - 12.2.3 Subversion 453
 - 12.3 Build Tools 454
 - 12.3.1 Make 455
 - 12.3.2 Ant 458
 - 12.3.3 Gradle 460
 - 12.4 Exceptions 462
 - 12.4.1 Handling Exceptions 464
 - 12.4.2 Custom Exceptions 466
 - 12.5 Documentation Tools 466
 - 12.5.1 Docstrings 467
 - 12.6 Program Diagnostics 468
 - 12.6.1 Debugging 468
 - Reference 470

13	Essential Knowledge: Data Stores	471
13.1	Data Persistence and Serialization	471
13.1.1	JSON	471
13.1.2	Pickle and Shelves in Python	473
13.1.3	Java Object Serialization	474
13.2	Hierarchical Data Format	476
13.2.1	Accessing HDF from Python Using PyTables	477
13.3	The Relational Database Model	478
13.3.1	The Relational Model	479
13.3.2	ACID	480
13.3.3	SQL Language	481
13.3.4	PostgreSQL, MySQL, and Other Database Solutions	489
13.3.5	Working with Databases: Shells and Programmatic APIs	490
13.4	NoSQL Databases	491
13.5	Memory Mapping	492
13.6	Notes	493
	References	493
14	Thoughts on System Design for Big Data	495
14.1	Where to Start?	495
14.2	The Big Picture	497
14.3	Load Balancing	499
14.4	Partitioning	501
14.5	Consistent Hashing	505
14.6	Scatter-Gather	506
14.7	Pre-Materialization	506
14.8	Blackboard	507
14.9	Pipelines	508
14.10	Redundancy, Recovery, and High Availability	510
14.10.1	Chaos Engineering	513
14.10.2	Fixing Forward	516
14.10.3	Rolling Back	516
14.11	Fault Tolerance	517
14.11.1	Retry Policies	517
14.11.2	Circuit Breakers	520
14.12	Offline, Near-Line, and Online Data Processing	521
14.13	Hot, Warm, and Cold Data Storage	521
14.14	The Cloud	522
14.14.1	Infrastructure-as-a-Service (IaaS)	523
14.14.2	Platform-as-a-Service (PaaS)	523
14.14.3	Functions-as-a-Service (FaaS)	524
14.15	Other Notable Cloud Services	524
14.15.1	Amazon Athena	525
14.15.2	Amazon DynamoDB	526

- 14.15.3 Amazon Elasticsearch Service (ES) 531
- 14.15.4 Amazon Elastic Map Reduce (EMR) 531
- 14.15.5 Amazon Glue 532
- 14.15.6 Amazon Kinesis..... 532
- 14.15.7 Amazon Redshift 533
- 14.15.8 Amazon Relational Database Service (RDS)..... 534
- 14.15.9 Amazon Simple Storage Service (S3) 534
- 14.16 Information Security 535
 - 14.16.1 Non-Repudiation..... 537
 - 14.16.2 Confidentiality 538
 - 14.16.3 Integrity 538
 - 14.16.4 Availability 539
 - 14.16.5 The STRIDE Threat Model 540
- 14.17 Notes..... 541
- References..... 541
- 15 Thoughts on Software Craftsmanship 543**
 - 15.1 Guiding Principles of Crafting Big Data Systems..... 544
 - 15.1.1 Sustainable Rapid Growth..... 545
 - 15.1.2 Balancing Rush Delivery and Craftsmanship 545
 - 15.1.3 Frequent Reassessment of Design Decisions 547
 - 15.1.4 The Incremental Cost-Effectiveness Ratio..... 548
 - 15.1.5 Repairing Broken Windows Frequently 549
 - 15.1.6 System Design Priorities 551
 - 15.2 Coding Style 554
 - 15.2.1 Naming 556
 - 15.2.2 Functions 558
 - 15.2.3 Comments 560
 - 15.2.4 Formatting..... 561
 - 15.2.5 API Design 564
 - 15.2.6 Error Handling 565
 - 15.2.7 Logging 568
 - 15.2.8 Tests 571
 - 15.3 Big Data Craftsmanship 571
 - 15.3.1 Metadata..... 572
 - 15.3.2 Discoverability 572
 - 15.3.3 Versioning 572
 - 15.3.4 Documentation..... 573
 - 15.3.5 Debuggability 574
 - 15.3.6 Quality..... 574
 - References..... 576