

# Contents

<b>Preface</b>	<b>xiii</b>
<b>Introduction</b>	<b>xvii</b>
<b>1 Why Did the Chicken Cross the Road?</b>	<b>1</b>
1.1 The Computer . . . . .	1
1.2 Turing Machine . . . . .	3
CT: ABSTRACT AWAY . . . . .	5
1.3 A Brief History of Computers . . . . .	5
1.4 Software . . . . .	7
1.5 Programming . . . . .	8
1.6 Syntax and Semantics . . . . .	9
CT: BEWARE OF SEMANTICS . . . . .	10
1.7 Flowcharts . . . . .	10
CT: READY FOR ALL CONTINGENCIES . . . . .	12
CT: FIRST THINGS FIRST . . . . .	12
CT: CHECK BEFORE PROCEEDING . . . . .	12
1.8 Algorithms . . . . .	13
CT: MAKE IT AN ALGORITHM . . . . .	14
CT: CONSIDER EXTREME CASES . . . . .	14
1.9 Pseudo Code . . . . .	15
CT: STEP BY STEP . . . . .	17
1.10 The Euclidean GCD Algorithm . . . . .	17
CT: APPLY DOMAIN KNOWLEDGE . . . . .	18
1.11 Goals and How to Get There . . . . .	18
CT: BREAK IT DOWN . . . . .	19
1.12 Road Crossing . . . . .	19
Exercises . . . . .	22
<b>2 Bits, Bytes, and Words</b>	<b>23</b>
2.1 Digital Computers . . . . .	23
2.2 Binary Numbers . . . . .	26
CT: MEANING OF SYMBOLS . . . . .	29
2.2.1 Numbers in Other Bases . . . . .	29
CT: EVALUATE DIFFERENT OPTIONS . . . . .	30

2.3	Positive and Negative Integers . . . . .	31
2.4	Modular Arithmetic . . . . .	33
	CT: MIND RESOURCE LIMITATIONS . . . . .	35
	CT: SYMBOLS CAN BE DECEIVING . . . . .	35
2.5	Base Conversion . . . . .	36
	CT: START FROM THE END . . . . .	37
2.6	Characters . . . . .	37
	2.6.1 US-ASCII . . . . .	37
	2.6.2 Unicode . . . . .	38
	CT: DATA CONTEXT . . . . .	40
2.7	Editing Text . . . . .	40
2.8	Data Output . . . . .	43
	CT: DELIVER THE MESSAGE . . . . .	44
	Exercises . . . . .	45
<b>3</b>	<b>True or False</b> . . . . .	<b>47</b>
3.1	Digital Electronic Circuits . . . . .	47
	CT: NOTICE THE LOGIC . . . . .	48
	CT: BOTTOM UP . . . . .	51
	CT: CREATE A VIRTUOUS CYCLE . . . . .	52
3.2	Boolean Algebra . . . . .	53
	3.2.1 Expressions and Laws . . . . .	53
	3.2.2 Universal Gates . . . . .	54
3.3	Decision Making . . . . .	54
	CT: LOGIC CHECKS . . . . .	56
	3.3.1 Conditions and Implications . . . . .	56
	CT: FOLLOW THE LOGIC . . . . .	58
3.4	Logic Applied to Bits . . . . .	58
	CT: COMBINE BASIC COMPONENTS . . . . .	61
3.5	Logic and Iteration . . . . .	61
	3.5.1 The <code>while</code> Loop . . . . .	61
	3.5.2 The <code>for</code> Loop . . . . .	62
	CT: PERFORM EVERYDAY PROGRAMMING . . . . .	66
	Exercises . . . . .	67
<b>4</b>	<b>Who Is the Master?</b> . . . . .	<b>69</b>
4.1	What Is an Operating System? . . . . .	69
4.2	Operating System Kernel . . . . .	70
	4.2.1 System Programs . . . . .	70
4.3	Open Source Software . . . . .	71
	CT: PROMOTE FREE AND OPEN . . . . .	72
4.4	Graphical User Interface . . . . .	72
4.5	Desktop Overview . . . . .	73
	4.5.1 Desktop Components . . . . .	73
	CT: KNOW YOUR ARENA . . . . .	75

4.6	Are You Talking to Me? . . . . .	75
	4.6.1 Input Focus . . . . .	75
	CT: PAY ATTENTION TO DETAILS . . . . .	76
	4.6.2 Event Handling . . . . .	76
4.7	Command-Line Interface . . . . .	77
	CT: MIND THE TRADE-OFF . . . . .	80
4.8	Files . . . . .	80
	4.8.1 File Content Types . . . . .	81
	4.8.2 File Tree . . . . .	81
	CT: LEARN FROM TREES . . . . .	83
	4.8.3 File Management and Access Control . . . . .	83
4.9	Processes . . . . .	84
	CT: KEEP IT IN CONTEXT . . . . .	85
	CT: CAPTURE THE STATE . . . . .	86
	4.9.1 Process Lifecycle . . . . .	86
	4.9.2 Process Address Space . . . . .	87
	4.9.3 Virtual Address Space Layout . . . . .	88
	4.9.4 Address Mapping . . . . .	89
	CT: TIMESHARING . . . . .	90
4.10	Managing Tasks . . . . .	90
4.11	Up and Running . . . . .	92
	CT: BETTER CONTROL BETTER SYSTEM . . . . .	92
	Exercises . . . . .	94
<b>5</b>	<b>Hello There!</b> . . . . .	<b>95</b>
5.1	What Is a Network? . . . . .	95
5.2	The Internet . . . . .	96
5.3	Local and Wide Area Networks . . . . .	98
5.4	Internet Architecture . . . . .	100
	CT: REDUNDANCY FOR SAFETY . . . . .	100
	CT: ONE AND ALL . . . . .	101
5.5	Wireless Networking . . . . .	102
5.6	Networking Protocols . . . . .	103
	CT: FOLLOW PROTOCOL . . . . .	104
5.7	IP Addresses . . . . .	105
5.8	Domain Names . . . . .	105
5.9	Client and Server . . . . .	106
	CT: INTEROPERATE . . . . .	107
5.10	Peer to Peer . . . . .	108
5.11	DNS Service . . . . .	109
	CT: INDIRECTION ADDS FLEXIBILITY . . . . .	110
5.12	DNS Servers and Resolvers . . . . .	111
	CT: DECENTRALIZE . . . . .	112
5.13	Domain Registration . . . . .	113
	5.13.1 Accessing Domain Registration Data . . . . .	113

5.14	Packet Switching . . . . .	114
5.15	Cloud Computing . . . . .	115
	CT: BACKUP IN THE CLOUD . . . . .	117
	Exercises . . . . .	119
<b>6</b>	<b>Home Sweet Homepage :-)</b>	<b>121</b>
6.1	What Is a Web Server? . . . . .	121
6.2	Web Browsers . . . . .	122
6.3	A Brief History of the Web . . . . .	123
6.4	URLs . . . . .	124
	CT: BE AWARE OF THE IMPLICIT CONTEXT . . . . .	126
	6.4.1 URL Encoding . . . . .	126
	CT: WEAR DIFFERENT HATS . . . . .	127
6.5	HTML and HTML5 . . . . .	128
	CT: MARK IT UP . . . . .	130
6.6	Webpage Styling . . . . .	130
6.7	Web Hosting . . . . .	130
	CT: REALLY USE YOUR WEBSITE . . . . .	131
	CT: BE CAREFUL WITH ONLINE INFORMATION . . . . .	131
6.8	Dynamic Generation of Webpages . . . . .	132
	6.8.1 Active Server Pages . . . . .	133
	6.8.2 Database Access . . . . .	134
6.9	Client-Side Scripting . . . . .	134
6.10	Hypertext Transfer Protocol . . . . .	135
	6.10.1 HTTP Caching . . . . .	137
	CT: CACHE FOR SPEED . . . . .	138
6.11	Website Development . . . . .	138
	CT: DEVELOP FOR USERS . . . . .	139
6.12	Web Search Engines . . . . .	139
	CT: GOOGLE IT . . . . .	140
	CT: BELIEVE IT OR NOT . . . . .	140
6.13	Web Services . . . . .	141
6.14	Standard Web Technologies . . . . .	142
	Exercises . . . . .	144
<b>7</b>	<b>Keeping It Safe</b>	<b>147</b>
7.1	Login . . . . .	148
	7.1.1 Website Login . . . . .	148
	CT: SAFEGUARD SECURITY REALMS . . . . .	149
	CT: PREVENT ILLICIT LOGIN . . . . .	151
7.2	HTTPS and SSL/TLS . . . . .	152
7.3	What Is a Digital Certificate? . . . . .	153
7.4	Cryptography . . . . .	155
	7.4.1 Symmetric Cryptosystems . . . . .	157
	CT: SECURE SENSITIVE FILES . . . . .	159

	CT: ADD SECURITY LAYERS . . . . .	159
7.5	Public-Key Cryptography . . . . .	160
	CT: BREAKTHROUGH . . . . .	160
	CT: BEWARE OF BUGS . . . . .	162
7.6	RSA Public-Key Algorithm . . . . .	162
7.7	Digital Signature . . . . .	164
	CT: SIGN DIGITALLY . . . . .	164
7.8	Message Digests . . . . .	164
7.9	Secure Email . . . . .	167
	7.9.1 Secure Email with Thunderbird . . . . .	167
	CT: FREE FROM SURVEILLANCE . . . . .	169
7.10	Security Attacks and Defenses . . . . .	170
	CT: ALL FOR ONE AND ONE FOR ALL . . . . .	171
	Exercises . . . . .	174
<b>8</b>	<b>Solve That Problem</b>	<b>175</b>
8.1	Solving Puzzles . . . . .	175
	8.1.1 Egg Frying . . . . .	175
	8.1.2 Liquid Measuring . . . . .	176
	8.1.3 A Magic Tray . . . . .	177
8.2	Sorting . . . . .	178
	8.2.1 Bubble Sort . . . . .	178
	8.2.2 Improved Bubble Sort . . . . .	180
	CT: CUT IT DOWN . . . . .	180
	CT: BUILD IT UP . . . . .	181
	CT: STEPWISE REFINEMENT . . . . .	181
	CT: VERSION 2.0 . . . . .	182
8.3	Recursion . . . . .	182
	CT: REMEMBER RECURSION . . . . .	183
	8.3.1 Quicksort . . . . .	184
8.4	Recursive Solution Formula . . . . .	186
	CT: APPLY THE RECURSION MAGIC . . . . .	186
8.5	Tower of Hanoi . . . . .	187
8.6	Eight Queens . . . . .	190
8.7	General Backtracking . . . . .	193
8.8	Tree Traversals . . . . .	194
	CT: FORM TREE STRUCTURES . . . . .	195
8.9	Complexity . . . . .	195
	CT: WEIGH SPEED VS. COMPLEXITY . . . . .	196
8.10	Heuristics . . . . .	197
	CT: DEVISE HEURISTICS . . . . .	197
	Exercises . . . . .	200
<b>9</b>	<b>Data Everywhere</b>	<b>201</b>
	CT: GARBAGE IN, GARBAGE OUT . . . . .	201

9.1	Digital Images . . . . .	202
9.1.1	Representing Color . . . . .	202
9.2	Raster Image Encoding . . . . .	204
9.2.1	Raster Image Formats . . . . .	204
	CT: SMALL IS BEAUTIFUL . . . . .	205
9.2.2	Vector Graphics . . . . .	205
9.2.3	Scalable Vector Graphics . . . . .	205
9.3	Audio and Video . . . . .	206
9.3.1	Digital Audio . . . . .	206
9.3.2	Audio Encoding Formats . . . . .	207
9.4	Digital Video . . . . .	208
9.4.1	Video Containers . . . . .	208
9.4.2	Video Codecs . . . . .	209
9.5	Format of Data and Files . . . . .	210
	CT: INTERPRETING DATA . . . . .	210
	CT: DATA IS APPLICATION DEPENDENT . . . . .	211
	CT: SAVE TREES WITH PDF . . . . .	211
9.6	Data Sharing . . . . .	212
9.7	Document Markup . . . . .	212
9.7.1	What Is XML? . . . . .	213
9.7.2	XML Document Format . . . . .	214
9.7.3	XML for News Syndication . . . . .	214
	CT: MARKUP FOR INTEROPERABILITY . . . . .	215
9.8	Data Compression . . . . .	216
	CT: COMPRESSION IS NOT ENCRYPTION . . . . .	217
9.8.1	LZ Deflation . . . . .	217
9.8.2	Huffman Code . . . . .	218
	CT: CUSTOMIZE FOR EFFICIENCY . . . . .	220
9.9	Data Structures . . . . .	220
	CT: SYNTHESIZE AND SIMPLIFY . . . . .	222
9.10	What Is a Database? . . . . .	222
9.10.1	Relational Databases . . . . .	222
9.10.2	SQL: Structured Query Language . . . . .	223
	CT: COMBINE WEB AND DATABASE . . . . .	224
9.10.3	Big Data . . . . .	225
	CT: DATA TO INSIGHT . . . . .	225
9.11	Protecting Personal Data . . . . .	225
	CT: GUARD PERSONAL DATA . . . . .	226
	Exercises . . . . .	227
<b>10</b>	<b>Get That App</b>	<b>229</b>
10.1	Key Programs . . . . .	230
	CT: REMIND YOURSELF . . . . .	231
	CT: INSTALL THAT APP . . . . .	231
10.2	Knowing Your Apps . . . . .	232

	CT: LEARN THAT APP . . . . .	233
	CT: NO APP, NO WAY . . . . .	233
10.3	Program Configuration and Customization . . . . .	234
	CT: CONFIGURE AND ENJOY . . . . .	234
10.4	Process Cooperation . . . . .	234
	CT: COORDINATE OR ELSE . . . . .	235
10.5	Machine Language Programs . . . . .	236
10.6	Assembly Language Programs . . . . .	237
10.7	High-Level Programs . . . . .	238
10.8	Compilers . . . . .	240
	CT: BOOTSTRAPPING . . . . .	241
10.9	Software Development . . . . .	242
10.10	Object-Oriented Programming . . . . .	242
	CT: COMPARTMENTALIZE . . . . .	242
	10.10.1 OOP Advantages . . . . .	243
	10.10.2 OOP Concepts . . . . .	244
	CT: EXPOSE ONLY THE INTERFACE . . . . .	244
10.11	Object-Oriented Design . . . . .	246
	Exercises . . . . .	247
	<b>Epilogue</b> . . . . .	<b>249</b>
	CT: REPROGRAM YOUR BRAIN . . . . .	249
	<b>Website and Interactive Demos</b> . . . . .	<b>251</b>
	<b>Bibliography</b> . . . . .	<b>253</b>
	<b>Index</b> . . . . .	<b>255</b>