

CONTENTS

	FOREWORD	xv
	<i>by Greg Wilson</i>	
	PREFACE	xvii
1	A REGULAR EXPRESSION MATCHER	1
	<i>by Brian Kernighan</i>	
	The Practice of Programming	2
	Implementation	3
	Discussion	4
	Alternatives	5
	Building on It	6
	Conclusion	8
2	SUBVERSION'S DELTA EDITOR: INTERFACE AS ONTOLOGY	11
	<i>by Karl Fogel</i>	
	Version Control and Tree Transformation	12
	Expressing Tree Differences	16
	The Delta Editor Interface	17
	But Is It Art?	23
	Abstraction As a Spectator Sport	25
	Conclusions	27
3	THE MOST BEAUTIFUL CODE I NEVER WROTE	29
	<i>by Jon Bentley</i>	
	The Most Beautiful Code I Ever Wrote	30
	More and More with Less and Less	31
	Perspective	36
	What Is Writing?	38
	Conclusion	39
	Acknowledgments	40
4	FINDING THINGS	41
	<i>by Tim Bray</i>	
	On Time	41
	Problem: Weblog Data	42
	Problem: Who Fetched What, When?	50
	Search in the Large	55
	Conclusion	57

5	CORRECT, BEAUTIFUL, FAST (IN THAT ORDER): LESSONS FROM DESIGNING XML VERIFIERS	59
	<i>by Elliotte Rusty Harold</i>	
	The Role of XML Validation	59
	The Problem	60
	Version 1: The Naïve Implementation	62
	Version 2: Imitating the BNF Grammar $O(N)$	63
	Version 3: First Optimization $O(\log N)$	64
	Version 4: Second Optimization: Don't Check Twice	66
	Version 5: Third Optimization $O(1)$	68
	Version 6: Fourth Optimization: Caching	72
	The Moral of the Story	74
6	FRAMEWORK FOR INTEGRATED TEST: BEAUTY THROUGH FRAGILITY	75
	<i>by Michael Feathers</i>	
	An Acceptance Testing Framework in Three Classes	76
	The Challenge of Framework Design	78
	An Open Framework	79
	How Simple Can an HTML Parser Be?	80
	Conclusion	83
7	BEAUTIFUL TESTS	85
	<i>by Alberto Savoia</i>	
	That Pesky Binary Search	87
	Introducing JUnit	89
	Nailing Binary Search	90
	Conclusion	102
8	ON-THE-FLY CODE GENERATION FOR IMAGE PROCESSING	105
	<i>by Charles Petzold</i>	
9	TOP DOWN OPERATOR PRECEDENCE	129
	<i>by Douglas Crockford</i>	
	JavaScript	130
	Symbol Table	131
	Tokens	132
	Precedence	133
	Expressions	134
	Infix Operators	134
	Prefix Operators	136
	Assignment Operators	137
	Constants	138
	Scope	138
	Statements	140

	Functions	143
	Array and Object Literals	144
	Things to Do and Think About	145
10	THE QUEST FOR AN ACCELERATED POPULATION COUNT <i>by Henry S. Warren, Jr.</i>	147
	Basic Methods	148
	Divide and Conquer	149
	Other Methods	151
	Sum and Difference of Population Counts of Two Words	152
	Comparing the Population Counts of Two Words	153
	Counting the 1-Bits in an Array	154
	Applications	158
11	SECURE COMMUNICATION: THE TECHNOLOGY OF FREEDOM <i>by Ashish Gulhati</i>	161
	The Heart of the Start	162
	Untangling the Complexity of Secure Messaging	163
	Usability Is the Key	165
	The Foundation	167
	The Test Suite	172
	The Functioning Prototype	172
	Clean Up, Plug In, Rock On...	173
	Hacking in the Himalayas	177
	The Invisible Hand Moves	182
	Speed Does Matter	184
	Communications Privacy for Individual Rights	185
	Hacking the Civilization	185
12	GROWING BEAUTIFUL CODE IN BIOPERL <i>by Lincoln Stein</i>	187
	BioPerl and the Bio::Graphics Module	188
	The Bio::Graphics Design Process	192
	Extending Bio::Graphics	210
	Conclusions and Lessons Learned	214
13	THE DESIGN OF THE GENE SORTER <i>by Jim Kent</i>	217
	The User Interface of the Gene Sorter	218
	Maintaining a Dialog with the User over the Web	219
	A Little Polymorphism Can Go a Long Way	221
	Filtering Down to Just the Relevant Genes	224
	Theory of Beautiful Code in the Large	225
	Conclusion	228

14	HOW ELEGANT CODE EVOLVES WITH HARDWARE: THE CASE OF GAUSSIAN ELIMINATION	229
	<i>by Jack Dongarra and Piotr Luszczek</i>	
	The Effects of Computer Architectures on Matrix Algorithms	230
	A Decompositional Approach	232
	A Simple Version	233
	LINPACK's DGEFA Subroutine	235
	LAPACK DGETRF	237
	Recursive LU	240
	ScaLAPACK PDGETRF	243
	Multithreading for Multi-Core Systems	247
	A Word About the Error Analysis and Operation Count	250
	Future Directions for Research	251
	Further Reading	252
15	THE LONG-TERM BENEFITS OF BEAUTIFUL DESIGN	253
	<i>by Adam Kolawa</i>	
	My Idea of Beautiful Code	253
	Introducing the CERN Library	254
	Outer Beauty	255
	Inner Beauty	261
	Conclusion	266
16	THE LINUX KERNEL DRIVER MODEL: THE BENEFITS OF WORKING TOGETHER	267
	<i>by Greg Kroah-Hartman</i>	
	Humble Beginnings	268
	Reduced to Even Smaller Bits	273
	Scaling Up to Thousands of Devices	276
	Small Objects Loosely Joined	277
17	ANOTHER LEVEL OF INDIRECTION	279
	<i>by Diomidis Spinellis</i>	
	From Code to Pointers	280
	From Function Arguments to Argument Pointers	282
	From Filesystems to Filesystem Layers	285
	From Code to a Domain-Specific Language	287
	Multiplexing and Demultiplexing	289
	Layers Forever?	290

18	PYTHON'S DICTIONARY IMPLEMENTATION: BEING ALL THINGS TO ALL PEOPLE	293
	<i>by Andrew Kuchling</i>	
	Inside the Dictionary	295
	Special Accommodations	296
	Collisions	298
	Resizing	299
	Iterations and Dynamic Changes	300
	Conclusion	301
	Acknowledgments	301
19	MULTIDIMENSIONAL ITERATORS IN NUMPY	303
	<i>by Travis E. Oliphant</i>	
	Key Challenges in N-Dimensional Array Operations	304
	Memory Models for an N-Dimensional Array	305
	NumPy Iterator Origins	307
	Iterator Design	307
	Iterator Interface	313
	Iterator Use	314
	Conclusion	318
20	A HIGHLY RELIABLE ENTERPRISE SYSTEM FOR NASA'S MARS ROVER MISSION	319
	<i>by Ronald Mak</i>	
	The Mission and the Collaborative Information Portal	320
	Mission Needs	321
	System Architecture	322
	Case Study: The Streamer Service	325
	Reliability	328
	Robustness	336
	Conclusion	338
21	ERP5: DESIGNING FOR MAXIMUM ADAPTABILITY	339
	<i>by Rogerio Atem de Carvalho and Rafael Monnerat</i>	
	General Goals of ERP	340
	ERP5	340
	The Underlying Zope Platform	342
	ERP5 Project Concepts	346
	Coding the ERP5 Project	347
	Conclusion	351

22	A SPOONFUL OF SEWAGE <i>by Bryan Cantrill</i>	353
23	DISTRIBUTED PROGRAMMING WITH MAPREDUCE <i>by Jeffrey Dean and Sanjay Ghemawat</i>	371
	A Motivating Example	371
	The MapReduce Programming Model	374
	Other MapReduce Examples	375
	A Distributed MapReduce Implementation	377
	Extensions to the Model	380
	Conclusion	381
	Further Reading	381
	Acknowledgments	382
	Appendix: Word Count Solution	382
24	BEAUTIFUL CONCURRENCY <i>by Simon Peyton Jones</i>	385
	A Simple Example: Bank Accounts	386
	Software Transactional Memory	388
	The Santa Claus Problem	396
	Reflections on Haskell	404
	Conclusion	404
	Acknowledgments	406
25	SYNTACTIC ABSTRACTION: THE SYNTAX-CASE EXPANDER <i>by R. Kent Dybuiğ</i>	407
	Brief Introduction to syntax-case	411
	Expansion Algorithm	413
	Example	425
	Conclusion	428
26	LABOR-SAVING ARCHITECTURE: AN OBJECT-ORIENTED FRAMEWORK FOR NETWORKED SOFTWARE <i>by William R. Otte and Douglas C. Schmidt</i>	429
	Sample Application: Logging Service	431
	Object-Oriented Design of the Logging Server Framework	433
	Implementing Sequential Logging Servers	439
	Implementing Concurrent Logging Servers	444
	Conclusion	450
27	INTEGRATING BUSINESS PARTNERS THE RESTFUL WAY <i>by Andrew Patzer</i>	451
	Project Background	452
	Exposing Services to External Clients	452
	Routing the Service Using the Factory Pattern	456
	Exchanging Data Using E-Business Protocols	457
	Conclusion	462

28	BEAUTIFUL DEBUGGING	463
	<i>by Andreas Zeller</i>	
	Debugging a Debugger	464
	A Systematic Process	466
	A Search Problem	467
	Finding the Failure Cause Automatically	468
	Delta Debugging	470
	Minimizing Input	472
	Hunting the Defect	473
	A Prototype Problem	475
	Conclusion	476
	Acknowledgments	476
	Further Reading	476
29	TREATING CODE AS AN ESSAY	477
	<i>by Yukihiko Matsumoto</i>	
30	WHEN A BUTTON IS ALL THAT CONNECTS YOU TO THE WORLD	483
	<i>by Arun Mehta</i>	
	Basic Design Model	484
	Input Interface	487
	Efficiency of the User Interface	500
	Download	500
	Future Directions	500
31	EMACSPeAK: THE COMPLETE AUDIO DESKTOP	503
	<i>by T. V. Raman</i>	
	Producing Spoken Output	504
	Speech-Enabling Emacs	505
	Painless Access to Online Information	516
	Summary	522
	Acknowledgments	525
32	CODE IN MOTION	527
	<i>by Laura Wingerd and Christopher Seiwald</i>	
	On Being “Bookish”	528
	Alike Looking Alike	529
	The Perils of Indentation	530
	Navigating Code	531
	The Tools We Use	532
	DiffMerge’s Checkered Past	534
	Conclusion	536
	Acknowledgments	536
	Further Reading	536

33	WRITING PROGRAMS FOR “THE BOOK”	539
	<i>by Brian Hayes</i>	
	The Nonroyal Road	540
	Warning to Parenthophobes	540
	Three in a Row	541
	The Slippery Slope	544
	The Triangle Inequality	545
	Meandering On	547
	“Duh!”—I Mean “Aha!”	548
	Conclusion	550
	Further Reading	550
	AFTERWORD	553
	<i>by Andy Oram</i>	
	CONTRIBUTORS	555
	INDEX	565