CONTENTS

	PREFACE	vii
1	CODE SHOULD BE EASY TO UNDERSTAND	1
	What Makes Code "Better"?	2
	The Fundamental Theorem of Readability	3
	Is Smaller Always Better?	3
	Does Time-Till-Understanding Conflict with Other Goals?	4
	The Hard Part	4
Part One	SURFACE-LEVEL IMPROVEMENTS	
2	PACKING INFORMATION INTO NAMES	7
	Choose Specific Words	8
	Avoid Generic Names Like tmp and retval	10
	Prefer Concrete Names over Abstract Names	13
	Attaching Extra Information to a Name	15
	How Long Should a Name Be?	18
	Use Name Formatting to Convey Meaning	20
	Summary	21
3	NAMES THAT CAN'T BE MISCONSTRUED	23
	Example: Filter()	24
	Example: Clip(text, length)	24
	Prefer min and max for (Inclusive) Limits	25
	Prefer first and last for Inclusive Ranges	26
	Prefer begin and end for Inclusive/Exclusive Ranges	26
	Naming Booleans	27
	Matching Expectations of Users	27
	Example: Evaluating Multiple Name Candidates	29
	Summary	31
4	AESTHETICS	33
	Why Do Aesthetics Matter?	34
	Rearrange Line Breaks to Be Consistent and Compact	35
	Use Methods to Clean Up Irregularity	37
	Use Column Alignment When Helpful	38
	Pick a Meaningful Order, and Use It Consistently	39
	Organize Declarations into Blocks	40
	Break Code into "Paragraphs"	41
	Personal Style versus Consistency	42
	Summary	43

5 KNOWING WHAT TO COMMENT 45
What NOT to Comment 47
Recording Your Thoughts 49
Put Yourself in the Reader's Shoes 51
Final Thoughts—Getting Over Writer's Block 56
Summary 57

6 MAKING COMMENTS PRECISE AND COMPACT 59
Keep Comments Compact 60
Avoid Ambiguous Pronouns 60
Polish Sloppy Sentences 61
Describe Function Behavior Precisely 61
Use Input/Output Examples That Illustrate Corner Cases 61
State the Intent of Your Code 62
"Named Function Parameter" Comments 63
Use Information-Dense Words 64
Summary 65

Part Two SIMPLIFYING LOOPS AND LOGIC

7 MAKING CONTROL FLOW EASY TO READ 69
The Order of Arguments in Conditionals 70
The Order of if/else Blocks 71
The ?: Conditional Expression (a.k.a. "Ternary Operator") 73
Avoid do/while Loops 74
Returning Early from a Function 75
The Infamous goto 76
Minimize Nesting 77
Can You Follow the Flow of Execution? 79
Summary 80

8 BREAKING DOWN GIANT EXPRESSIONS 83
Explaining Variables 84
Summary Variables 84
Using De Morgan's Laws 85
Abusing Short-Circuit Logic 86
Example: Wrestling with Complicated Logic 86
Breaking Down Giant Statements 89
Another Creative Way to Simplify Expressions 90
Summary 90

9 VARIABLES AND READABILITY 93
Eliminating Variables 94
Shrink the Scope of Your Variables 97
Prefer Write-Once Variables 103
A Final Example 104
Summary 106

Part Three REORGANIZING YOUR CODE

10 EXTRACTING UNRELATED SUBPROBLEMS 109
Introductory Example: findClosestLocation() 110
Pure Utility Code 111
Other General-Purpose Code 112
Create a Lot of General-Purpose Code 114
Project-Specific Functionality 115
Simplifying an Existing Interface 116
Reshaping an Interface to Your Needs 117
Taking Things Too Far 117
Summary 118

11 ONE TASK AT A TIME 121
Tasks Can Be Small 123
Extracting Values from an Object 124
A Larger Example 128
Summary 130

12 TURNING THOUGHTS INTO CODE 131
Describing Logic Clearly 132
Knowing Your Libraries Helps 133
Applying This Method to Larger Problems 134
Summary 137

13 WRITING LESS CODE 139
Don't Bother Implementing That Feature—You Won't Need It 140
Question and Break Down Your Requirements 140
Keeping Your Codebase Small 142
Be Familiar with the Libraries Around You 143
Example: Using Unix Tools Instead of Coding 144
Summary 145

Part Four SELECTED TOPICS

14 TESTING AND READABILITY 149
Make Tests Easy to Read and Maintain 150
What's Wrong with This Test? 150
Making This Test More Readable 151
Making Error Messages Readable 154
Choosing Good Test Inputs 156
Naming Test Functions 158
What Was Wrong with That Test? 159
Test-Friendly Development 160
Going Too Far 162
Summary 162

15 DESIGNING AND IMPLEMENTING A "MINUTE/HOUR COUNTER" 165
The Problem 166
Defining the Class Interface 166

Attempt 1: A Naive Solution 169
Attempt 2: Conveyor Belt Design 171
Attempt 3: A Time-Bucketed Design 174
Comparing the Three Solutions 179
Summary 179

A FURTHER READING 181

INDEX 185