Table of Contents

Introduction 1

"What Will Understanding C Do for Me?" 1
About This Here Dummies Approach 2
How to Work the Examples in This Book 2
Foolish Assumptions 3
Icons Used in This Book 3
What's New with This Edition? 4
Final Thots 4

Part 1: Introduction to C Programming 7

Chapter 1: Up from the Primordial C 9

An Extremely Short and Cheap History of the C Language 9
The C Development Cycle 11
From Text File to Program 11
The source code (text file) 12
Creating the GOODBYE.C source code file 13
The compiler and the linker 14
Compiling GOODBYE.C 15
Running the final result 16
Save It! Compile and Link It! Run It! 16

Chapter 2: C of Sorrow, C of Woe 19

The Required Woes of Editing and Recompiling 19
Reediting your source code file 20
Recompiling (or the C equivalent of the "do-over") 21
Dealing with the Heartbreak of Errors 22
Yikes! An error! But, before you shoot yourself. 22
The autopsy 23
Repairing the malodorous program 24
Now try this error! 26

Chapter 3: C Straight 29

The Big Picture 29
C Language Pieces' Parts 30
The C Language Itself — the Keywords 32
Other C Language Components 34
Pop Quiz! 35
The Helpful RULES Program 36
The importance of being `\n` 36
Breaking up lines`\` is easy to do 37

Chapter 4: C What I/O 39

Introduce Yourself to Mr. Computer 39
Compiling WHORU.C 40
The reward 41
More on `printf()` 41
Printing funky text 42
Escape from `printf()`! 44
The f means "formatted" 46
A bit of justification 47
`scanf` Is Pronounced "Scan-Eff" 49
Putting `scanf` together 49
The miracle of `scanf()` 51
Experimentation time! 52

Chapter 5: To C or Not to C 55

Adding Comments 55
A big, hairy program with comments 56
Why are comments necessary? 58
Comment Styles of the Nerdy and Not-Quite-Yet-Nerdy 58
Bizarr-o comments 59
C++ comments 60
Using Comments to Disable 61
The Perils of "Nested" Comments 62

Chapter 6: C More I/O with `gets()` and `puts()` 65

The More I Want, the More I `gets()` 65
Another completely rude program example 66
And now, the bad news about `gets()` 67
The Virtues of `puts()` 67
Another silly command-prompt program 68
`puts()` and `gets()` in action 68
More insults 69
`puts()` can print variables 70

Part II: Run and Scream from Variables and Math 73

Chapter 7: A + B = C 75

The Ever-Changing Variable 75
Strings change 76
Running the KITTY 77
Welcome to the Cold World of Numeric Variables 77
Hello, integer 78
Using an integer variable in the Methuselah program 79

Assigning values to numeric variables 80
Entering numeric values from the keyboard 81
The `atoi()` function 81
So how old is this Methuselah guy, anyway? 83
You and Mr. Wrinkles 85
A Wee Bit o' Math 86
Basic mathematical symbols 86
How much longer do you have to live to break the Methuselah record? 88
Bonus modification on the final Methuselah program! 90
The direct result 91

Chapter 8: Charting Unknown Cs with Variables 93

Cussing, Discussing, and Declaring Variables 93
"Why must I declare a variable?" 94
Variable names verboten and not 95
Presetting variable values 96
The old random-sampler variable program 98
Maybe you want to chance two pints? 99
Multiple declarations 100
Constants and Variables 101
Dreaming up and defining constants 101
The handy shortcut 102
The `#define` directive 104
Real, live constant variables 106

Chapter 9: How to C Numbers 107

There Are Numbers, and Then There Are Numbers 107
Numbers in C 108
Why use integers? Why not just make every number floating-point? 110
Integer types (short, long, wide, fat, and so on) 110
Signed or unsigned, or "Would you like a minus sign with that, Sir?" 111
How to Make a Number Float 113
"Hey, Carl, let's write a floating-point number program!" 114
The E notation stuff 116
Bigger than the Float, It's a Double! 118
Formatting Your Zeroes and Decimal Places 119

Chapter 10: Cook That C Variable Charred, Please 121

The Other Kind of Variable Type, the `char` 121
Single-character variables 122
`Char` in action 123
Stuffing characters into character variables 124

Reading and Writing Single Characters 125
The `getchar()` function 126
The `putchar()` function 127
Character Variables As Values 128

Part III: Giving Your Programs the Ability to Run Amok 131

Chapter 11: C More Math and the Sacred Order of Precedence ...133

An All-Too-Brief Review of the Basic C Mathematical Operators 133
The old "how tall are you" program 135
Unethical alterations to the old "how tall are you" program 136
The Delicate Art of Incrementation (Or, "Just Add One to It") 137
Unhappily incrementing your weight 138
Bonus program! (One that may even have a purpose in life) 140
The Sacred Order of Precedence 141
A problem from the pages of the dentistry final exam 141
What's up, Sally? 142
The confounding magic-pellets problem 144
Using parentheses to mess up the order of precedence 145

Chapter 12: C the Mighty `if` Command 147

If Only. 147
The computer-genie program example 148
The `if` keyword, up close and impersonal 150
A question of formatting the `if` statement 154
The final solution to the income-tax problem 155
If It Isn't True, What Else? 157
Covering all the possibilities with `else` 158
The `if` format with `else` 159
The strange case of `else-if` and even more decisions 160
Bonus program! The really, really smart genie 163

Chapter 13: What If `C==C`? 165

The World of `if` without Values 165
Which is greater: S or T, $ or –? 166
The problem with `getchar()` 168
Fixing GREATER.C to easily read standard input 170
"Can I get `getchar()` to read only one character?" 171
Meanwhile, back to the GREATER problem 171
Another, bolder example 173
Using the `if` Keyword to Compare Two Strings 174

Chapter 14: Iffy C Logic 175

Exposing Flaws in logic 175
If, And, Or, But 177
A solution (but not the best one) 177
A better solution, using logic 178
The `if` command's logical friends 180
A logical `AND` program for you 183

Chapter 15: C You Again 185

For Going Loopy 185
Repetitive redundancy, I don't mind 187
For doing things over and over, use the `for` keyword 188
Tearing through OUCH.C a step at a time 190
Having fun whilst counting to 100 192
I'm Bustin' Outta Here! 193
At last — the handy ASCII program 193
Beware of infinite loops! 195
Breaking out of a loop 197
The `break` keyword 198

Chapter 16: C the Loop, C the Loop++ 201

The Art of Incrementation 201
Cryptic C operator symbols, Volume I: The `inc` operator (++) 202
Another look at the LARDO.C program 203
The Mysterious Practice of Decrementation 204
O, to count backward 205
How counting backward fits into the `for` loop 206
Cryptic C operator symbols, Volume II: The `dec` operator (`--`) 207
A final improvement to OLLYOLLY.C 208
More Incrementation Madness 209
Leaping loops! 210
Counting to 1,000 by fives 211
Cryptic C operator symbols, Volume III:
The madness continues 211
The answers 213

Chapter 17: C You in a While Loop 215

The Lowdown on `while` Loops 215
Whiling away the hours 216
The `while` keyword (a formal introduction) 218
Deciding between a `while` loop and a `for` loop 219
Replacing those unsightly `for(;;)` loops
with elegant `while` loops 220
C from the inside out 222
Not to Beat a Dead Horse or Anything. 223

Chapter 18: Do C While You Sleep 225

The Down-Low on Upside-Down `do-while` Loops 225
The devil made me `do-while` it! 226
`do-while` details 227
A flaw in the COUNTDWN.C program 228
The always kosher number-checking `do-while` loop 229
Nested Loops and Other Bird-Brained Concepts 231
Adding a tense, dramatic delay to the COUNTDWN.C program 231
The nitty GRID.C of nested loops 234
`Break` the Brave and `Continue` the Fool 235
Please continue. 236
The `continue` keyword 237

Chapter 19: Switch Case, or, From 'C' to Shining 'c' 239

The Sneaky `switch-case` Loops 239
The `switch-case` Solution to the LOBBY Program 241
The Old `switch-case` Trick 243
The Special Relationship between `while` and `switch-case` 248

Part IV: C Level.....251

Chapter 20: Writing That First Function 253

Meet Mr. Function 253
A silly example you don't have to type 254
A potentially redundant program in need of a function 255
The noble `jerk()` function 256
How the `jerk()` function works in BIGJERK2.C 257
Prototyping Your Functions 258
Prototypical prototyping problems 259
A sneaky way to avoid prototyping problems 260
The Tao of Functions 262
The function format 262
How to name your functions 263

Chapter 21: Contending with Variables in Functions 265

Bombs Away with the BOMBER Program! 265
Will the dual variable BOMBER.C program bomb? 267
Adding some important tension 267
How We Can All Share and Love with Global Variables 269
Making a global variable 270
An example of a global variable in a real, live program 271

Chapter 22: Functions That Actually Funct 275

Marching a Value Off to a Function 275
How to send a value to a function 276
An example (and it's about time!) 277
Avoiding variable confusion (must reading) 279
Sending More than One Value to a Function 280
Functions That Return Stuff 282
Something for your troubles 282
Finally, the computer tells you how smart it thinks you are 284
Return to sender with the `return` keyword 285
Now you can understand the `main()` function 287
Give that human a bonus! 288
No Need to Bother with This C Language Trivia If You're in a Hurry 289

Chapter 23: The Stuff That Comes First 293

Please Don't Leave Me Out! 294
Say! Aren't you the `#include` construction? 294
What's up with STDIO.H? 297
Writing your own dot-H file 298
A final warning about header files 300
What the `#defines` Are Up To 302
Avoiding the Topic of Macros 303

Chapter 24: The `printf()` Chapter 305

A Quick Review of `printf()` 305
The Old Displaying-Text-with-`printf()` Routine 306
The `printf()` Escape Sequences 306
The `printf()` escape-sequence testing program deluxe 307
Putting PRINTFUN to the test 308
The Complex `printf()` Format 310
The `printf()` Conversion Characters 311

Chapter 25: Math Madness! 313

More on Math 313
Taking your math problems to a higher power 314
Putting `pow()` into use 315
Rooting out the root 317
Strange Math? You Got It! 319
Something Really Odd to End Your Day 320
The perils of using `a++` 320
Oh, and the same thing applies to a `--` 322
Reflections on the strange `++a` phenomenon 322

Chapter 26: The Old Random-Number Function 325

On Being Random.....325
Using the `rand()` function 326
Planting a random-number seed 328
Randoming up the RANDOM program.....329
Streamlining the randomizer 331
The Diabolical Dr. Modulus.....333
Rolling the Dice with the Final RANDOM Program 335

Part V: Part of Tens 337

Chapter 27: Ten More Things You Need to Know about the C Language 339

Arrays 339
Strings.....340
Structures.....341
Pointers 343
Linked Lists.....343
Binary Operators 344
Interacting with the Command Line 345
Disk Access 345
Interacting with the Operating System 345
Building Big Programs 346

Chapter 28: Ten Tips for the Budding Programmer 347

Use the Command-Line History 347
Keep Your Editor Open in Another Window.....348
Use a Context-Colored Text Editor 348
Know the Line-Number Commands in Your Editor.....349
Keep a Command Prompt Window Open If You're Using the IDE.....350
Know a Few Handy Command-Prompt Commands 350
Carefully Name Your Variables 351
Know Your Post- and Pre-Incrementing and Decrementing Riddles.....351
Breaking Out of a Loop.....352

Chapter 29: Ten Ways to Solve Your Own Programming Problems 353

Work on One Thing at a Time 354
Break Up Your Code.....354
Simplify.....355
Talk through the Program 355
Set Breakpoints 356

Monitor Your Variables....356
Document Your Work....356
Use Debugging Tools....357
Use a C Optimizer....357
Read More Books!....358

Appendix A: The Stuff You Need to Know before You Read All the Other Stuff in This Book....359

Setting Things Up....359
The C language compiler....360
The place to put your stuff....361
Making Programs....363
Finding your `learn` directory or folder....363
Running an editor....364
Compiling and linking....365

Appendix B: ASCII Table....371

Index....377