Making Everything Easier!™

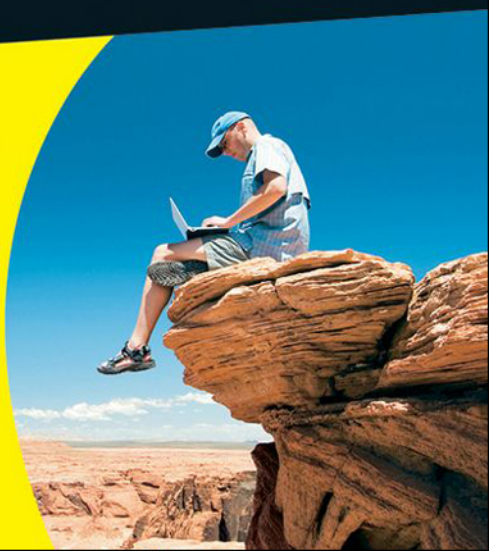
Beginning Programming with C++

FOR DUMMIES®

Learn to:

- Think like a programmer and understand how C++ works
- Create programs and get bugs out of your code
- Master basic development concepts and techniques in C++

Find source code from the book and the Code::Blocks C++ compiler on the companion CD-ROM



Stephen R. Davis
Author of C++ For Dummies

Beginning Programming with C++ FOR DUMMIES®

by Stephen R. Davis

Wiley Publishing, Inc.

Disclaimer: This eBook does not include ancillary media that was packaged with the printed version of the book.

Beginning Programming with C++ For Dummies®
Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Trademarks: Wiley, the Wiley Publishing logo, For Dummies, the Dummies Man logo, A Reference for the Rest of Us!, The Dummies Way, Dummies Daily, The Fun and Easy Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2010930969

ISBN: 978-0-470-61797-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Introduction 1

Part 1: Let's Get Started 7

Chapter 1: What Is a Program? 9
Chapter 2: Installing Code::Blocks 21
Chapter 3: Writing Your First Program 33

Part 11: Writing a Program: Decisions, Decisions 45

Chapter 4: Integer Expressions 47
Chapter 5: Character Expressions 59
Chapter 6: if I Could Make My Own Decisions 69
Chapter 7: Switching Paths 81
Chapter 8: Debugging Your Programs, Part I 89

Part 111: Becoming a Functional Programmer 97

Chapter 9: while Running in Circles 99
Chapter 10: Looping for the Fun of It 109
Chapter 11: Functions, I Declare! 117
Chapter 12: Dividing Programs into Modules 129
Chapter 13: Debugging Your Programs, Part 2 139

Part 1V: Data Structures 149

Chapter 14: Other Numerical Variable Types 151
Chapter 15: Arrays 165
Chapter 16: Arrays with Character 173
Chapter 17: Pointing the Way to C++ Pointers 187
Chapter 18: Taking a Second Look at C++ Pointers 203
Chapter 19: Programming with Class 223
Chapter 20: Debugging Your Programs, Part 3 235

Part V: Object-Oriented Programming 251

Chapter 21: What Is Object-Oriented Programming? 253
Chapter 22: Structured Play: Making Classes Do Things 259
Chapter 23: Pointers to Objects 269
Chapter 24: Do Not Disturb: Protected Members 281
Chapter 25: Getting Objects Off to a Good Start 289

Chapter 26: Making Constructive Arguments 303
Chapter 27: Coping with the Copy Constructor 323

Part VI: Advanced Strokes 333

Chapter 28: Inheriting a Class 335
Chapter 29: Are Virtual Functions for Real? 343
Chapter 30: Overloading Assignment Operators 355
Chapter 31: Performing Streaming I/O 363
Chapter 32: I Take Exception! 387

Part VII: The Part of Tens 397

Chapter 33: Ten Ways to Avoid Bugs 399
Chapter 34: Ten Features Not Covered in This Book 405

Appendix: About the CD 411

Index 415

Table of Contents

Introduction 1

About Beginning Programming with C++ For Dummies 1
Foolish Assumptions 2
Conventions Used in This Book 2
What You Don't Have to Read 3
How This Book Is Organized 3
Part I: Let's Get Started 3
Part II: Writing a Program: Decisions, Decisions 4
Part III: Becoming a Functional Programmer 4
Part IV: Data Structures 4
Part V: Object-Oriented Programming 4
Part VI: Advanced Strokes 4
Part VII: The Part of Tens 5
The CD-ROM Appendix 5
Icons Used in This Book 5
Where to Go from Here 6

Part 1: Let's Get Started 7

Chapter 1: What Is a Program? 9

How Does My Son Differ from a Computer? 9
Programming a "Human Computer" 11
The algorithm 11
The Tire Changing Language 12
The program 13
Computer processors 16
Computer Languages 17
High level languages 18
The C++ language 19

Chapter 2: Installing Code::Blocks 21

Reviewing the Compilation Process 21
Installing Code::Blocks 23
Testing the Code::Blocks Installation 25
Creating the project 27
Testing your default project 30

Chapter 3: Writing Your First Program 33

Creating a New Project 33
Entering Your Program 35
Building the Program 38

Finding What Could Go Wrong 38
Misspelled commands 38
Missing semicolon 40
Using the Enclosed CD-ROM 41
Running the Program 42
How the Program Works 42
The template 42
The Conversion program 44

Part II: Writing a Program: Decisions, Decisions 45

Chapter 4: Integer Expressions 47

Declaring Variables 47
Variable names 48
Assigning a value to a variable 49
Initializing a variable at declaration 49
Integer Constants 50
Expressions 51
Binary operators 51
Decomposing compound expressions 53
Unary Operators 54
The Special Assignment Operators 56

Chapter 5: Character Expressions 59

Defining Character Variables 59
Encoding characters 60
Example of character encoding 63
Encoding Strings of Characters 65
Special Character Constants 65

Chapter 6: if I Could Make My Own Decisions 69

The if Statement 69
Comparison operators 70
Say “No” to “No braces” 72
What else Is There? 73
Nesting if Statements 75
Compound Conditional Expressions 78

Chapter 7: Switching Paths 81

Controlling Flow with the switch Statement 81
Control Fell Through: Did I break It? 84
Implementing an Example Calculator with the switch Statement 85

Chapter 8: Debugging Your Programs, Part I. 89

Identifying Types of Errors 89
Avoiding Introducing Errors 90
Coding with style 90
Establishing variable naming conventions 91
Finding the First Error with a Little Help 92
Finding the Run-Time Error 93
Formulating test data 93
Executing the test cases 94
Seeing what's going on in your program 95

Part III: Becoming a Functional Programmer 97

Chapter 9: while Running in Circles 99

Creating a while Loop 99
Breaking out of the Middle of a Loop 102
Nested Loops 105

Chapter 10: Looping for the Fun of It 109

The for Parts of Every Loop 109
Looking at an Example 111
Getting More Done with the Comma Operator 113

Chapter 11: Functions, I Declare! 117

Breaking Your Problem Down into Functions 117
Understanding How Functions Are Useful 118
Writing and Using a Function 119
Returning things 120
Reviewing an example 121
Passing Arguments to Functions 123
Function with arguments 124
Functions with multiple arguments 125
Exposing main() 125
Defining Function Prototype Declarations 127

Chapter 12: Dividing Programs into Modules 129

Breaking Programs Apart 129
Breaking Up Isn't That Hard to Do 130
Creating Factorial.cpp 131
Creating an #include file 133
Including #include files 134
Creating main.cpp 136
Building the result 137

Using the Standard C++ Library 137
Variable Scope 137

Chapter 13: Debugging Your Programs, Part 2 139

Debugging a Dys-Functional Program 139
Performing unit level testing 141
Outfitting a function for testing 143
Returning to unit test 146

Part IV: Data Structures 149

Chapter 14: Other Numerical Variable Types 151

The Limitations of Integers in C++ 151
Integer round-off 151
Limited range 152
A Type That "doubles" as a Real Number 153
Solving the truncation problem 153
When an integer is not an integer 154
Discovering the limits of double 155
Variable Size — the "long" and "short" of It 158
How far do numbers range? 159
Types of Constants 160
Passing Different Types to Functions 161
Overloading function names 162
Mixed mode overloading 162

Chapter 15: Arrays 165

What Is an Array? 165
Declaring an Array 166
Indexing into an Array 167
Looking at an Example 168
Initializing an Array 171

Chapter 16: Arrays with Character 173

The ASCII-Zero Character Array 173
Declaring and Initializing an ASCIIZ Array 174
Looking at an Example 175
Looking at a More Detailed Example 177
Foiling hackers 181
Do I Really Have to Do All That Work? 182

Chapter 17: Pointing the Way to C++ Pointers 187

What's a Pointer? 187
Declaring a Pointer 188

Passing Arguments to a Function 190
Passing arguments by value 190
Passing arguments by reference 193
Putting it together 195
Playing with Heaps of Memory 197
Do you really need a new keyword? 197
Don't forget to clean up after yourself 198
Looking at an example 199

Chapter 18: Taking a Second Look at C++ Pointers 203

Pointers and Arrays 203
Operations on pointers 203
Pointer addition versus indexing into an array 205
Using the pointer increment operator 208
Why bother with array pointers? 210
Operations on Different Pointer Types 212
Constant Nags 212
Differences Between Pointers and Arrays 214
My main() Arguments 214
Arrays of pointers 215
Arrays of arguments 216

Chapter 19: Programming with Class 223

Grouping Data 223
The Class 224
The Object 225
Arrays of Objects 226
Looking at an Example 227

Chapter 20: Debugging Your Programs, Part 3 235

A New Approach to Debugging 235
The solution 236
Entomology for Dummies 236
Starting the debugger 239
Navigating through a program with the debugger 241
Fixing the (first) bug 245
Finding and fixing the second bug 246

Part V: Object-Oriented Programming 251

Chapter 21: What Is Object-Oriented Programming? 253

Abstraction and Microwave Ovens 253
Functional nachos 254
Object-oriented nachos 255
Classification and Microwave Ovens 256
Why Build Objects This Way? 256
Self-Contained Classes 257

Chapter 22: Structured Play: Making Classes Do Things 259

Activating Our Objects 259
Creating a Member Function 261
Defining a member function 261
Naming class members 262
Calling a member function 263
Accessing other members from within a member function 264
Keeping a Member Function after Class 266
Overloading Member Functions 267

Chapter 23: Pointers to Objects 269

Pointers to Objects 269
Arrow syntax 270
Calling all member functions 271
Passing Objects to Functions 271
Calling a function with an object value 271
Calling a function with an object pointer 272
Looking at an example 274
Allocating Objects off the Heap 278

Chapter 24: Do Not Disturb: Protected Members 281

Protecting Members 281
Why you need protected members 282
Making members protected 282
So what? 285
Who Needs Friends Anyway? 286

Chapter 25: Getting Objects Off to a Good Start 289

The Constructor 289
Limitations on constructors 291
Can I see an example? 292
Constructing data members 294
Destructors 297
Looking at an example 297
Destructing data members 300

Chapter 26: Making Constructive Arguments 303

Constructors with Arguments 303
Looking at an example 304
Overloading the Constructor 307
The Default default Constructor 312
Constructing Data Members 313
Initializing data members with the default constructor 314
Initializing data members with a different constructor 315
Looking at an example 318
New with C++ 2009 321

Chapter 27: Coping with the Copy Constructor 323

Copying an Object 323
The default copy constructor 324
Looking at an example 325
Creating a Copy Constructor 327
Avoiding Copies 330

Part VI: Advanced Strokes 333

Chapter 28: Inheriting a Class 335

Advantages of Inheritance 336
Learning the lingo 337
Implementing Inheritance in C++ 337
Looking at an example 338
Having a HAS_A Relationship 342

Chapter 29: Are Virtual Functions for Real? 343

Overriding Member Functions 343
Early binding 344
Ambiguous case 346
Enter late binding 348
When Is Virtual Not? 351
Virtual Considerations 352

Chapter 30: Overloading Assignment Operators 355

Overloading an Operator 355
Overloading the Assignment Operator Is Critical 356
Looking at an Example 358
Writing Your Own (or Not) 361

Chapter 31: Performing Streaming I/O 363

How Stream I/O Works 363
Stream Input/Output 365
Creating an input object 365
Creating an output object 366
Open modes 367
What is binary mode? 368
Hey, file, what state are you in? 369
Other Member Functions of the fstream Classes 373
Reading and writing streams directly 375
Controlling format 378
What's up with endl? 380
Manipulating Manipulators 380
Using the stringstream Classes 382

Chapter 32: I Take Exception! 387

The Exception Mechanism 387
Examining the exception mechanism in detail 390
Special considerations for throwing 391
Creating a Custom Exception Class 392
Restrictions on exception classes 395

Part VII: The Part of Tens..... 397

Chapter 33: Ten Ways to Avoid Bugs 399

Enable All Warnings and Error Messages.....399
Adopt a Clear and Consistent Coding Style 400
Comment the Code While You Write It.....401
Single-Step Every Path in the Debugger at Least Once.....401
Limit the Visibility 402
Keep Track of Heap Memory.....402
Zero Out Pointers after Deleting What They Point To.....403
Use Exceptions to Handle Errors.....403
Declare Destructors Virtual 403
Provide a Copy Constructor and Overloaded Assignment Operator ... 404

Chapter 34: Ten Features Not Covered in This Book 405

The goto Command.....405
The Ternary Operator.....406
Binary Logic 407
Enumerated Types 407
Namespaces 407
Pure Virtual Functions 408
The string Class 408
Multiple Inheritance.....409
Templates and the Standard Template Library.....409
The 2009 C++ Standard 410

Appendix: About the CD..... 411

Index..... 415

Introduction

Welcome to *Beginning Programming with C++ For Dummies.* This book is intended for the reader who wants to learn to program.

Somehow over the years, programming has become associated with mathematics and logic calculus and other complicated things. I never quite understood that. Programming is a skill like writing advertising or drawing or photography. It does require the ability to think a problem through, but I've known some really good programmers who had zero math skills. Some people are naturally good at it and pick it up quickly, others not so good and not so quick. Nevertheless, anyone with enough patience and "stick-to-itiveness" can learn to program a computer. Even me.

About Beginning Programming with C++ For Dummies

Learning to program necessarily means learning a programming language. This book is based upon the C++ programming language. A Windows version of the suggested compiler is included on the CD-ROM accompanying this book. Macintosh and Linux versions are available for download at `www.codeblocks.org`. (Don't worry: I include step-by-step instructions for how to install the package and build your first program in the book.)

The goal of this book is to teach you the basics of programming in C++, not to inundate you with every detail of the C++ programming language. At the end of this book, you will be able to write a reasonably sophisticated program in C++. You will also be in a position to quickly grasp a number of other similar languages, such as Java and C#.NET.

In this book, you will discover what a program is, how it works, plus how to do the following:

- Install the CodeBlocks C++ compiler and use it to build a program
- Create and evaluate expressions
- Direct the flow of control through your program

- Create data structures that better model the real world
- Define and use C++ pointers
- Manipulate character strings to generate the output the way you want to see it
- Write to and read from files

Foolish Assumptions

I try to make very few assumptions in this book about the reader, but I do assume the following:

- **You have a computer.** Most readers will have computers that run Windows; however, the programs in this book run equally well on Windows, Macintosh, Linux, and Unix. In fact, since C++ is a standardized language, these programs should run on any computer that has a C++ compiler.
- **You know the basics of how to use your computer.** For example, I assume that you know how to run a program, copy a file, create a folder, and so on.
- **You know how to navigate through menus.** I include lots of instructions like "Click on File and then Open." If you can follow that instruction, then you're good to go.
- **You are new to programming.** I don't assume that you know anything about programming. Heck, I don't even assume that you know what programming is.

Conventions Used in This Book

To help you navigate this book as efficiently as possible, I use a few conventions:

- C++ terms are in monofont typeface, `like this`.
- New terms are emphasized with *italics* (and defined).
- Numbered steps that you need to follow and characters you need to type are set in **bold**.

What You Don't Have to Read

I encourage you to read one part of the book; then put the book away and play for a while before moving to the next part. The book is organized so that by the end of each part, you have mastered enough new material to go out and write programs.

I'd like to add the following advice:

- If you already know what programming is but nothing about C++, you can skip Chapter 1.
- I recommend that you use the CodeBlocks compiler that comes with the book, even if you want to use a different C++ compiler after you finish the book. However, if you insist and don't want to use CodeBlocks, you can skip Chapter 2.
- Skim through Chapter 3 if you've already done a little computer programming.
- Start concentrating at Chapter 4, even if you have experience with other languages such as BASIC.
- You can stop reading after Chapter 20 if you're starting to feel saturated. Chapter 21 opens up the new topic of object-oriented programming — you don't want to take that on until you feel really comfortable with what you've learned so far.
- You can skip any of the TechnicalStuff icons.

How This Book Is Organized

Beginning Programming with C++ For Dummies is split into seven parts. You don't have to read it sequentially, and you don't even have to read all the sections in any particular chapter. You can use the Table of Contents and the Index to find the information you need and quickly get your answer. In this section, I briefly describe what you'll find in each part.

Part 1: Let's Get Started

This part describes what programs are and how they work. Using a fictitious tire-changing computer, I take you through several algorithms for removing a tire from a car to give you a feel for how programs work. You'll also get CodeBlocks up and running on your computer before leaving this part.

Part II: Writing a Program: Decisions, Decisions

This part introduces you to the basics of programming with C++. You will find out how to declare integer variables and how to write simple expressions. You'll even discover how to make decisions within a program, but you won't be much of an expert by the time you finish this part.

Part III: Becoming a Functional Programmer

Here you learn how to direct the flow of control within your programs. You'll find out how to loop, how to break your code into modules (and why), and how to build these separate modules back into a single program. At the end of this part, you'll be able to write real programs that actually solve problems.

Part IV: Data Structures

This part expands your knowledge of data types. Earlier sections of the book are limited to integers; in this part, you work with characters, decimals, and arrays; and you even get to define your own types. Finally, this is the part where you master the most dreaded topic, the C++ pointer.

Part V: Object-Oriented Programming

This is where you expand your knowledge into object-oriented techniques, the stuff that differentiates C++ from its predecessors, most notably C. (Don't worry if you don't know what object-oriented programming is — you aren't supposed to yet.) You'll want to be comfortable with the material in Parts I through IV before jumping into this part, but you'll be a much stronger programmer by the time you finish it.

Part VI: Advanced Strokes

This is a collection of topics that are important but that didn't fit in the earlier parts. For example, here's where I discuss how to create, read to, and write from files.

Part VII: The Part of Tens

This part includes lists of what to do (and what not to do) when programming to avoid creating bugs needlessly. In addition, this part includes some advice about what topics to study next, should you decide to expand your knowledge of C++.

The CD-ROM Appendix

This part describes what's on the enclosed CD-ROM and how to install it.

Icons Used in This Book

What's a *Dummies* book without icons pointing you in the direction of really great information that's sure to help you along your way? In this section, I briefly describe each icon I use in this book.

The Tip icon points out helpful information that is likely to make your job easier.

This icon marks a generally interesting and useful fact — something that you might want to remember for later use. I also use this icon to remind you of some fact that you may have skipped over in an earlier chapter.

The Warning icon highlights lurking danger. With this icon, I'm telling you to pay attention and proceed with caution.

When you see this icon, you know that there's techie stuff nearby. If you're not feeling very techie, you can skip this info.

This icon denotes the programs that are included on this book's CD-ROM.

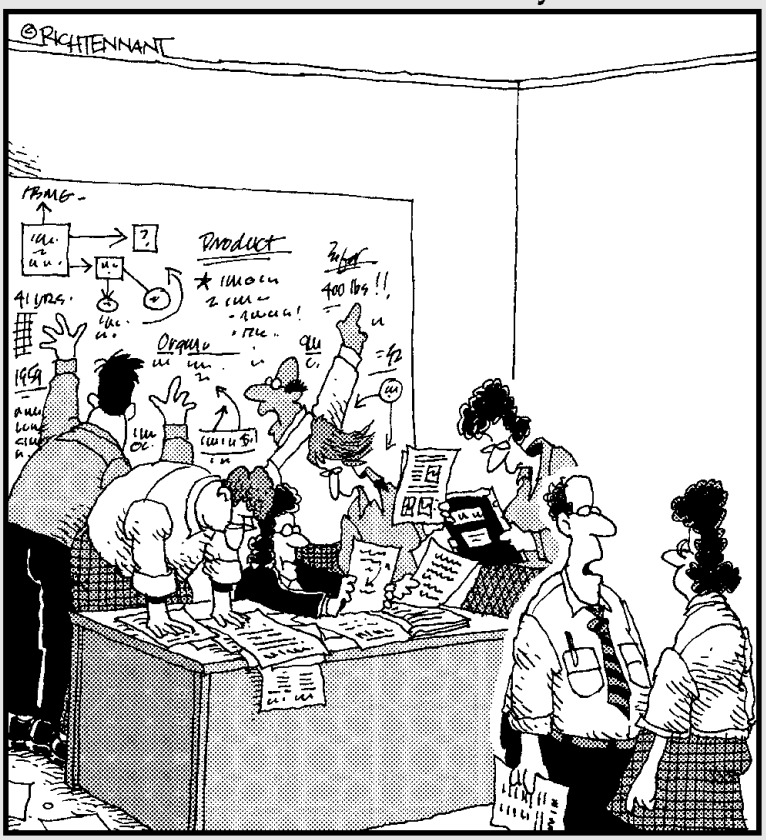
Where to Go from Here

You can find a set of errata and Frequently Asked Questions for this and all my books at `www.stephendavis.com`. You will also find a link to my e-mail address there. Feel free to send me your questions and comments (that's how I learn). It's through reader input that these books can improve.

Now you've stalled long enough, it's time to turn to Chapter 1 and start discovering how to program!

Part I
Let's Get Started

The 5th Wave By Rich Tennant



"We're outsourcing everything but our core competency. Once we find out what that is, we'll begin the outsourcing process."

In this part . . .

You will learn what it means to program a computer. You will also get your first taste of programming — I take you through the steps to enter, build, and execute your first program. It will all be a bit mysterious in this part, but things will clear up soon, I promise.

Chapter 1

What Is a Program?

In This Chapter

- Understanding programs
- Writing your first "program"
- Looking at computer languages

In this chapter, you will learn what a program is and what it means to write a program. You'll practice on a Human Computer. You'll then see some program snippets written for a real computer. Finally, you'll see your first code snippet written in C++.

Up until now all of the programs running on your computer were written by someone else. Very soon now, that won't be true anymore. You will be joining the ranks of the few, the proud: the programmers.

How Does My Son Differ from a Computer?

A computer is an amazingly fast but incredibly stupid machine. A computer can do anything you tell it (within reason), but it does *exactly* what it's told — nothing more and nothing less.

In this respect, a computer is almost the exact opposite of a human: humans respond intuitively. When I was learning a second language, I learned that it wasn't enough to understand what was being said — it's just as important and considerably more difficult to understand what was left unsaid. This is information that the speaker shares with the listener through common experience or education — things that don't need to be said.

For example, I say things to my son like, "Wash the dishes" (for all the good it does me). This seems like clear enough instructions, but the vast majority of the information contained in that sentence is implied and unspoken.

Let's assume that my son knows what dishes are and that dirty dishes are normally in the sink. But what about knives and forks? After all, I only said dishes, I didn't say anything about eating utensils, and don't even get me started on glassware. And did I mean wash them manually, or is it okay to load them up into the dishwasher to be washed, rinsed, and dried automatically?

But the fact is, "Wash the dishes" is sufficient instruction for my son. He can decompose that sentence and combine it with information that we both share, including an extensive working knowledge of dirty dishes, to come up with a meaningful understanding of what I want him to do — whether he does it or not is a different story. I would guess that he can perform all the mental gymnastics necessary to understand that sentence in about the same amount of time that it takes me to say it — about 1 to 2 seconds.

A computer can't make heads or tails out of something as vague as "wash the dishes." You have to tell the computer exactly what to do with each different type of dish, how to wash a fork, versus a spoon, versus a cup. When does the program stop washing a dish (that is, how does it know when a dish is clean)? When does it stop washing (that is, how does it know when it's finished)?

My son has gobs of memory — it isn't clear exactly how much memory a normal human has, but it's boat loads. Unfortunately, human memory is fuzzy. For example, witnesses to crimes are notoriously bad at recalling details even a short time after the event. Two witnesses to the same event often disagree radically on what transpired.

Computers also have gobs of memory, and that's very good. Once stored, a computer can retrieve a fact as often as you like without change. As expensive as memory was back in the early 1980s, the original IBM PC had only 16K (that's 16 thousand bytes). This could be expanded to a whopping 64K. Compare this with the 1GB to 3GB of main storage available in most computers today (1GB is *one billion bytes*).

As expensive as memory was, however, the IBM PC included extra memory chips and decoding hardware to detect a memory failure. If a memory chip went bad, this circuitry was sure to find it and report it before the program went haywire. This so-called Parity Memory was no longer offered after only a few years, and as far as I know, it is unavailable today except in specific applications where extreme reliability is required — because the memory boards almost never fail.

On the other hand, humans are very good at certain types of processing that computers do poorly, if at all. For example, humans are very good at pulling the meaning out of a sentence garbled by large amounts of background noise. By contrast, digital cell phones have the infuriating habit of just going silent whenever the noise level gets above a built-in threshold.

Programming a "Human Computer"

Before I dive into showing you how to write programs for computer consumption, I start by showing you a program to guide human behavior so that you can better see what you're up against. Writing a program to guide a human is much easier than writing programs for computer hardware because we have a lot of familiarity with and understanding of humans and how they work (I assume). We also share a common human language to start with. But to make things fair, assume that the human computer has been instructed to be particularly literal — so the program will have to be very specific. Our guinea pig computer intends to take each instruction quite literally.

The problem I have chosen is to instruct our human computer in the changing of a flat tire.

The algorithm

The instructions for changing a flat tire are straightforward and go something like the following:

1. Raise the car.
2. Remove the lug nuts that affix the faulty tire to the car.
3. Remove the tire.
4. Mount the new tire.
5. Install the lug nuts.
6. Lower the car.

(I know that technically the lug nuts hold the wheel onto the car and not the tire, but that distinction isn't important here. I use the terms "wheel" and "tire" synonymously in this discussion.)

sample content of Beginning Programming with C++ For Dummies

- Philosophy in History: Essays on the Historiography of Philosophy (Ideas in Context) pdf
- **Basic Technical Analysis of Financial Markets: A Modern Approach (Perspectives in Busines Culture) book**
- Psychiatric Nursing (6th Edition) book
- read Pfalz Scout Aces of World War 1
- Learn You a Haskell for Great Good!: A Beginner's Guide pdf, azw (kindle)

- http://conexdxb.com/library/Philosophy-in-History--Essays-on-the-Historiography-of-Philosophy--Ideas-in-Context-.pdf
- http://schrolf.de/books/Basic-Technical-Analysis-of-Financial-Markets--A-Modern-Approach--Perspectives-in-Business-Culture-.pdf
- http://anvilpr.com/library/Le-Sermon-Sur-La-Chute-De-Rome.pdf
- http://www.wybohaas.com/freebooks/The-No-Nonsense-Guide-to-Animal-Rights--No-Nonsense-Guides-.pdf
- http://growingsomeroots.com/ebooks/Cool-Down--Getting-Further-by-Going-Slower.pdf