

***Beginning
Programming***
FOR
DUMMIES®
4TH EDITION

Wallace Wang



Wiley Publishing, Inc.

***Beginning
Programming***
FOR
DUMMIES®
4TH EDITION

Wallace Wang



Wiley Publishing, Inc.

Beginning Programming For Dummies®, 4th Edition

Published by

Wiley Publishing, Inc.

111 River Street

Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2007 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 Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, 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, 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 800-762-2974, 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: 2006932692

ISBN-13: 978-0-470-08870-8

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

4O/RW/RQ/QW/IN



About the Author

Wallace Wang is one of many carbon-based life forms currently populating the planet Earth. He began his working career by going to college and getting a “good” job — only to find that a college education never guaranteed you a “good” job and most “good” jobs actually stink.

So faced with the prospect of spending the rest of his waking life in a caffeine-induced stupor coupled with shots of alcohol and gambling to dull the frustration of a dead-end job, he decided to pursue one of many dreams by becoming a writer. First starting out writing magazine articles for a local San Diego computer magazine, he soon graduated to writing for national publications and book publishers as well.

After nearly 20 years of writing full-time, he’s still pursuing a variety of different dreams besides continuing book and magazine writing. He’s still bouncing around comedy clubs around Michigan, Las Vegas, and San Diego, performing stand-up comedy to anyone sober enough to listen.

He’s also branched away from computer books by teaming up with coauthors to write non-computer related books, most notably *Breaking Into Acting For Dummies* with Larry Garrison. (Thanks go to Ben Affleck for carrying around a copy and getting his picture taken with the book, which appeared in many major magazines including *People Magazine* and *The New York Post*.)

In his latest mad venture to avoid having to work in an ordinary 9–5 job, he’s also teamed up with three other comedians (Rick Gene, Wes Sample, and Justin Davis) to create, produce, and host a radio show called “Keeping It Weird,” currently (at least at the time of this writing) airing on 103.7 Free FM in San Diego.

By the time you read this, the author may be off pursuing something entirely different. Whatever he may be doing at the time, it should at least make for interesting stories to tell his grandchildren about one day.

Dedication

This book is dedicated to all the wonderful people I've met along the path of life, including . . .

Cassandra (my wife), Jordan (my son), and Bo, Scraps, Tasha, and Nuit (our cats).

Lily Carnie, the only person I know who can truly see both sides of the story.

All the friendly folks I've met while performing at the Riviera Comedy Club, located at the Riviera Hotel & Casino in Las Vegas: Steve Schirripa (who also appears on the HBO show, *The Sopranos*, which you can read about at www.hbo.com/sopranos), Don Learned, Bob Zany, Gerry Bednob, Bruce Clark, Darrell Joyce, and Kip Addotta. The next time you're visiting Las Vegas, drop by the Riviera and watch a comedy show. Then dump some money in a slot machine on the way out to ensure that the Riviera Hotel & Casino continues making enough money to keep its comedy club open.

Thanks also go to Roger Feeny at the Ann Arbor Comedy Showcase in Ann Arbor, Michigan; Russ Rivas at Laff's in Albuquerque, New Mexico; Pat Wilson at Mesquite, Nevada; and Joe Jarred at Primm and Pahrump, Nevada for running some of the friendliest comedy clubs around the country.

Final thanks must also go to Leo (the man, the myth, the legend) Fontaine, Chris (the Zooman) Clobber, Rick Gene, Wes Sample, Justin Davis, and Dante (who gets excited just to see his name in a book).

Author's Acknowledgments

If it wasn't for Bill Gladstone at Waterside Productions, I might still be staring off into space in a cubicle somewhere, working in a dead-end job, wondering what could have been.

Additional thanks go to Allen Wyatt for making sure that everything in this book is accurate, along with Rebecca Senninger and Virginia Sanders for making the process of writing a book always painless, easy, and often fun.

Final acknowledgements go to Cassandra (my wife) for putting up with multiple computers that (from her point of view) seem to spontaneously appear and disappear around the house at random. Each time a computer disappears, a more advanced model appears that promises more speed and hard drive space, but still never seems to have more speed or as much room as the previous computer model that it replaced.

Publisher's Acknowledgments

We're proud of this book; please send us your comments through our online registration form located at www.dummies.com/register/.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Project Editor: Rebecca Senninger

(Previous Edition: Andrea C. Boucher)

Acquisitions Editor: Katie Feltman

Copy Editor: Virginia Sanders

Technical Editor: Allen Wyatt

Editorial Manager: Leah Cameron

Media Development Specialists: Angela Denny,
Kate Jenkins, Steven Kudirka, Kit Malone

Media Development Coordinator:
Laura Atkinson

Media Project Supervisor: Laura Moss

Media Development Manager:
Laura VanWinkle

Editorial Assistant: Amanda Foxworth

Sr. Editorial Assistant: Cherie Case

Cartoons: Rich Tennant
(www.the5thwave.com)

Composition Services

Project Coordinator: Michael Kruzil

Layout and Graphics: Claudia Bell,
Carl Byers, Brooke Graczyk, Denny Hager,
Joyce Haughey, Stephanie D. Jumper,
Barbara Moore, Barry Offringa,
Laura Pence, Alicia South

Proofreaders: Laura Albert, Techbooks

Indexer: Steve Rath

Anniversary Logo Design: Richard Pacifico

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Joyce Pepple, Acquisitions Director

Composition Services

Gerry Fahey, Vice President of Production Services

Debbie Stailey, Director of Composition Services

Contents at a Glance

<i>Introduction</i>	<i>1</i>
<i>Part I: Programming a Computer.....</i>	<i>7</i>
Chapter 1: Learning Computer Programming for the First Time	9
Chapter 2: All about Programming Languages	19
Chapter 3: How to Write a Program	37
Chapter 4: The Tools of a Computer Programmer	49
<i>Part II: The Building Blocks of Programming</i>	<i>63</i>
Chapter 5: Getting Started.....	65
Chapter 6: The Structure of a Computer Program	75
Chapter 7: Variables, Constants, and Comments	91
Chapter 8: Crunching Numbers and Playing with Strings.....	103
Chapter 9: Making Decisions with Branching Statements	119
Chapter 10: Repeating Yourself with Loops	143
Chapter 11: Dividing a Program into Subprograms	157
Chapter 12: Storing Stuff in Arrays.....	175
Chapter 13: Playing with Object-Oriented Programming.....	195
<i>Part III: Advanced Programming Topics.....</i>	<i>213</i>
Chapter 14: Sorting and Searching Algorithms	215
Chapter 15: Debugging Programs	239
Chapter 16: Optimizing Your Code.....	249
Chapter 17: Creating a User Interface.....	259
<i>Part IV: Internet Programming.....</i>	<i>279</i>
Chapter 18: Playing with HTML	281
Chapter 19: Making Interactive Web Pages with JavaScript	309
Chapter 20: Using Java Applets on Web Pages	321

<i>Part V: The Part of Tens</i>	<i>327</i>
Chapter 21: Ten Additional Programming Resources.....	329
Chapter 22: Ten Cool Programming Careers.....	341
<i>Appendix A: Common Loop and Branching Structures</i>	<i>351</i>
<i>Appendix B: Free Language Compilers and Interpreters.....</i>	<i>355</i>
<i>Appendix C: Common Programming Terms.....</i>	<i>357</i>
<i>Appendix D: Installing the CD Compilers</i>	<i>365</i>
<i>Index</i>	<i>369</i>
<i>Wiley Publishing, Inc. End-User License Agreement</i>	<i>389</i>

Table of Contents

Introduction 1

Who Should Buy This Book	1
How This Book Is Organized	3
Part I: Programming a Computer	3
Part II: The Building Blocks of Programming	3
Part III: Advanced Programming Topics	3
Part IV: Internet Programming	4
Part V: The Part of Tens	4
How to Use This Book	4
Foolish assumptions	4
Icons used in this book	5

Part 1: Programming a Computer 7

Chapter 1: Learning Computer Programming for the First Time 9

Why Learn Computer Programming?	9
How Does a Computer Program Work?	13
Programming is problem-solving	14
Programming isn't difficult; it's just time-consuming	15
What Do I Need to Know to Program a Computer?	17

Chapter 2: All about Programming Languages 19

Why So Many Different Programming Languages?	19
The joy of assembly language	20
C: The portable assembler	22
High-level programming languages	24
Rapid application development (RAD) programming languages	26
Database programming languages	29
Scripting programming languages	30
Web page programming languages	32
So What's the Best Programming Language to Learn?	34

Chapter 3: How to Write a Program 37

Before You Write Your Program	37
The problem	38
The program's users	38

The target computer	39
Your programming skill	39
The Technical Details of Writing a Program	40
Prototyping	40
Choosing a programming language.....	41
Defining how the program should work.....	42
The Life Cycle of a Typical Program	44
The development cycle	45
The maintenance cycle.....	46
The upgrade cycle.....	46

Chapter 4: The Tools of a Computer Programmer 49

Writing Programs in an Editor	50
Using a Compiler or an Interpreter.....	51
Compilers	52
Interpreters	53
P-code: A combination compiler and interpreter	54
So what do I use?	56
Squashing Bugs with a Debugger	56
Writing a Help File.....	59
Creating an Installation Program	60

Part 11: The Building Blocks of Programming.....63

Chapter 5: Getting Started 65

Learning BASIC	66
Getting to know Liberty BASIC	66
Getting to know REALbasic	67
Learning C++	69
Learning Revolution.....	71

Chapter 6: The Structure of a Computer Program 75

Designing a Program.....	76
Spaghetti programming	77
The three parts of every program.....	79
The building blocks of programming.....	83
Dividing and Conquering with Subprograms	85
Creating libraries of subprograms	87
Dividing programs into objects	88

Chapter 7: Variables, Constants, and Comments	.91
Creating Variables	92
Naming variables	93
Defining the data type of a variable	94
Assigning a value to a variable	97
Using Constants	99
Commenting Your Code	100
Chapter 8: Crunching Numbers and Playing with Strings	.103
Adding, Subtracting, Dividing, and Multiplying	103
Working with precedence	105
Using parentheses	107
Using Built-In Math Functions	108
Manipulating Strings	111
Declaring variables as strings	111
Smashing strings together	112
Converting Strings into Numbers (And Vice Versa)	113
Converting strings into numbers	114
Converting a number into a string	117
Chapter 9: Making Decisions with Branching Statements	.119
Using Boolean Expressions	119
Storing Boolean expressions in variables	121
Using variables in Boolean expressions	122
Using Boolean operators	123
Exploring IF-THEN Statements	128
Using IF-THEN-ELSE Statements	129
Working with SELECT CASE Statements	132
Checking a range of values	138
Checking a relational operator	140
Chapter 10: Repeating Yourself with Loops	.143
Creating Loops	143
Looping a Fixed Number of Times	144
Counting with different numbers	147
Counting in increments	147
Counting backward	149
Using a WHILE Loop	150
The UNTIL Loop Variation	152
Watching Out for Endless Loops	154
Endless loops #1: Failing to modify the Boolean expression inside the loop	154
Endless loops #2: Failing to initialize a Boolean expression outside the loop	155

Chapter 11: Dividing a Program into Subprograms157

Creating a Subprogram.....	158
Creating a Liberty BASIC subprogram.....	158
Creating a REALbasic subprogram.....	160
Creating a C++ subprogram.....	163
Creating a Revolution subprogram	164
Running Subprograms	165
Passing parameters in Liberty BASIC	165
Passing parameters in REALbasic	168
Passing parameters in C++	169
Passing parameters in Revolution	170
Using Functions	170
Creating a Liberty BASIC function.....	171
Creating a REALbasic function	172
Creating a C++ function.....	173
Creating a Revolution function	174

Chapter 12: Storing Stuff in Arrays175

Creating an Array	175
Using arrays in Liberty BASIC.....	176
Using arrays in REALbasic	178
Using arrays in C++.....	179
Using arrays in Revolution	181
Creating Dynamic Arrays	182
Using dynamic arrays in Liberty BASIC.....	182
Using dynamic arrays in REALbasic	184
Using dynamic arrays in C++	187
Using dynamic arrays in Revolution.....	188
Creating a Multi-Dimensional Array.....	189
Using two-dimensional arrays in Liberty BASIC.....	190
Using two-dimensional arrays in REALbasic	191
Using two-dimensional arrays in C++.....	192
Using two-dimensional arrays in Revolution	193

Chapter 13: Playing with Object-Oriented Programming195

Encapsulation: Isolating Data	196
Creating a class.....	198
Writing an object's methods	199
Creating an object from a class	200
Inheritance: Reusing Code	204
Polymorphism: Rewriting Code	208

Part III: Advanced Programming Topics213**Chapter 14: Sorting and Searching Algorithms215**

Insertion Sort.....	216
Bubble Sort	220
Shell Sort	223
Quicksort.....	227
Choosing a Sorting Algorithm	231
Using a Built-In Sorting Algorithm	231
Searching Algorithms	232
Searching sequentially.....	233
Performing a binary search.....	235
Picking a Searching Algorithm.....	238

Chapter 15: Debugging Programs239

Anatomy of a Computer Bug.....	240
Syntax Errors	240
Run-Time Errors	242
Fun with Logic Errors	243
Stepping line by line.....	244
Tracing through your program.....	245
Watching your variables.....	246

Chapter 16: Optimizing Your Code249

Choosing the Right Data Structure	249
Choosing the Right Algorithm	251
Fine-Tuning the Source Code.....	252
Putting the condition most likely to be false first.....	252
Putting the condition most likely to be true first.....	252
Don't run a FOR-NEXT loop needlessly	254
Cleaning out your loops	254
Using the correct data types.....	256
Using built-in commands whenever possible	256
Using a Faster Language.....	257
Optimizing Your Compiler	258

Chapter 17: Creating a User Interface259

Designing a Window.....	261
Putting Pull-Down Menus in a Window	263
Displaying Controls.....	264
Command buttons.....	266
Check boxes and radio buttons.....	268
List boxes and combo boxes	270
Text boxes and labels	276

Part IV: Internet Programming279**Chapter 18: Playing with HTML281**

Grasping the Basics of HTML	282
Grasping the most important HTML tags.....	283
Creating a header and title.....	283
Defining the bulk of your Web page	284
Adding comments	284
Defining Text with Tags	285
Making a heading.....	285
Defining a paragraph.....	286
Highlighting a quote.....	287
Adding emphasis to text.....	288
Using Tag Attributes	289
Aligning text	289
Playing with colors.....	289
Coloring your hyperlinks.....	290
Making a List.....	291
Unordered lists	291
Ordered lists	292
Definition lists	293
Creating Hyperlinks	295
Making external hyperlinks.....	295
Making internal hyperlinks	296
Linking to a specific spot on a Web page	296
Displaying Graphics	297
Putting a picture on a Web page.....	298
Adding a background picture	299
Creating a User Interface on a Form	299
Handling events	300
Creating a text box.....	301
Creating a command button	302
Creating a check box.....	303
Creating a radio button.....	306

Chapter 19: Making Interactive Web Pages with JavaScript309

Understanding the Basics of JavaScript.....	310
Displaying text	311
Creating variables.....	312
Making dialog boxes	313
Playing with Functions	315
Opening and Closing a Window.....	318
Opening a window	318
Defining a window's appearance	318
Closing a window.....	319

Chapter 20: Using Java Applets on Web Pages 321

How Java Applets Work.....	321
Adding a Java Applet to a Web Page.....	323
Defining the size of a Java applet window	323
Aligning the location of a Java applet window	324
Defining space around a Java applet	325

Part V: The Part of Tens 327**Chapter 21: Ten Additional Programming Resources 329**

Developing Windows Programs	330
Developing Macintosh Programs	331
Developing Linux Programs	332
Using a Database Language	333
Clarion.....	333
PowerBuilder	334
FileMaker	334
SQL	334
Shopping by Mail Order	335
Getting Your Hands on Source Code	335
Joining a Local User Group	336
Frequenting Usenet Newsgroups	337
Playing Core Wars	337
Programming a Battling Robot	338
Toying with Lego Mindstorms	339

Chapter 22: Ten Cool Programming Careers 341

Programming Computer Games for Fun and Profit	341
Creating Computer Animation.....	343
Making (And Breaking) Encryption	344
Internet Programming	345
Fighting Malware	346
Hacking for Hire.....	346
Participating in an Open-Source Project	347
Niche-Market Programming.....	348
Teaching Others about Computers.....	349
Selling Your Own Software	349

Appendix A: Common Loop and Branching Structures... 351

Common Loop Structures	351
Common Branching Structures	352

<i>Appendix B: Free Language Compilers and Interpreters</i>	<i>355</i>
<i>Appendix C: Common Programming Terms</i>	<i>357</i>
<i>Appendix D: Installing the CD Compilers</i>	<i>365</i>
System Requirements	365
Using the CD	366
If You Have Problems (Of the CD Kind).....	367
<i>Index</i>	<i>369</i>
<i>Wiley Publishing, Inc. End-User License Agreement.....</i>	<i>389</i>

Introduction



A*nyone* can learn to program a computer. Computer programming doesn't require a high IQ or an innate proficiency in advanced mathematics. Computer programming just requires a desire to learn and the patience never to give up.

Programming is a skill like rock climbing, tap dancing, and pole vaulting. Some people are naturally better than others, but anyone can get better with regular practice. That's why so many kids become programming wizards at such an early age. These kids aren't necessarily brilliant; they're just willing to put in the time to learn a new skill, and they're not afraid of failing because they know that failure is nothing more than a part of learning.

If you ever dreamed about writing your own programs, rest assured that you can. Programming can be lots of fun, but it can also be frustrating, annoying, and time-consuming. That's why Wiley publishes this particular book — to help you discover how to program a computer with minimum inconvenience and maximum enjoyment.

Whether you want to pick up computer programming for fun, to start a new career, or to help make your current job easier, consider this book your personal guide through the sometimes scary — and initially intimidating — world of computer programming.

Although this book won't turn you into a programming wizard overnight, it can teach you enough about programming to help you understand how programming works, what the strengths and weaknesses of different programming languages are, and how you can get started writing programs all by yourself.

Who Should Buy This Book

You should buy this book if you want to learn how computer programming works without getting bogged down in the technical details of a particular programming language. When you understand how computer programming works, you'll better understand how to use a specific programming language with cryptic names like C++ or Java. But you should buy this book if you especially want to know any of the following:

- ✓ How computer programs work
- ✓ The common parts of every computer programming language
- ✓ How to write programs for multiple platforms such as Windows, Mac OS X, or Linux
- ✓ Whether to write your next computer program by using Visual Basic, C++, Perl, SmallTalk, C#, or some other programming language

Like any skill, you can learn programming only by practicing it. To help you get hands-on experience, the CD enclosed with this book includes trial versions of four language compilers so you can practice writing programs on any computer that runs Windows, Mac OS X, or Linux.

The three main languages you learn in this book are BASIC, C++, and a scripting language called Revolution.

BASIC is specially designed to introduce beginners to programming, so you can practice writing BASIC programs in two programming languages: Liberty BASIC and REALbasic.

Liberty BASIC represents the BASIC language in its purest and simplest form so you can understand the concepts of programming without getting lost in the technical details. When you understand how BASIC works, you can study REALbasic to see a version of the BASIC language that includes advanced programming features similar to more powerful languages such as C++.

This book also provides examples in C++, which is the most popular programming language in use today. If you want to write programs professionally, you must at least become familiar with the way C++ works.

You also learn a nontraditional programming language called Revolution, which uses English-like sentences to control your computer. Scripting languages like Revolution are designed to be easy to write and understand. They also provide commands capable of solving complicated problems more easily than traditional programming languages like BASIC or C++.

In addition, the Revolution programming language is based on AppleScript, which is a programming language used to automate a Mac OS X computer, so after you're familiar with Revolution, you also know most of the AppleScript programming language.

By learning BASIC, C++, and Revolution, you're exposed to three different programming languages, styles, and approaches to solving problems so you can better understand the advantages and limitations of any programming language. Then you can choose the best programming language for your needs.

How This Book Is Organized

To help you find what you need quickly, this book consists of five parts, and each part covers a certain topic about computer programming. Whenever you need help, just flip through the book, find the part that covers the topic you're looking for, and then keep the book at your side as you get back to work.

Part I: Programming a Computer

If computer programming seems a mysterious arcane science, relax. This part of the book demystifies all the common myths about computer programming, shows you exactly how computer programs work, and explains why programming isn't as difficult as many people think.

This part also shows you how programming has evolved, why so many different programming languages exist, and how programming follows easy-to-remember principles so you can start programming your own computer right away.

Part II: The Building Blocks of Programming

Although literally thousands of different programming languages are available for you to learn, every programming language tends to work in similar ways. So in this part of the book, you learn the basic building blocks of writing and creating a program regardless of the particular programming language you use.

To help you understand the building blocks of programming, each chapter provides plenty of examples in different programming languages so you can see how they accomplish the same task. You can also try out the examples on your own computer.

Part III: Advanced Programming Topics

After you master the basics of writing a program, you need to worry about making your program work efficiently, eliminating problems, and designing a user interface so other people will know how to use it. In this part of the book, you learn how programmers fine-tune their software (and what the consequences might be if they don't).

Part IV: Internet Programming

The Internet is fast becoming an integral part of the computer world, so this part of the book introduces you to the basics of various Internet languages, including HTML (which designs the appearance of Web pages), JavaScript, and Java.

In this part, you also see how other people create cool Web pages that look good and can display forms and respond to users. You can use this information to create Web sites that interact with users.

Part V: The Part of Tens

To help gently guide you toward writing your own programs for money, this part of the book provides information about programming jobs you might want to pursue and how to find more tools and source code to help you learn more about programming all by yourself.

How to Use This Book

This book is meant to show you the basics of computer programming without bogging you down with the technical details of any particular programming language. Typed code often looks like chicken scratches or the random characters that a monkey might type if left alone with a keyboard. So you can use this book as a tutorial (to show you how programming works) and as a reference (to help refresh your memory for understanding different programming techniques).

Ideally, you want to use this book along with your computer. Read some of the book and then try what you just read on your computer so that you can see with your own eyes how programming works.

Foolish assumptions

To get the most out of this book, you need access to a computer (because trying to understand computer programming without a computer is like trying to learn to drive without a car). To take full advantage of this book, you need a computer running Microsoft Windows, Mac OS X, or Linux.

Icons used in this book

Icons highlight useful tips, important information to remember, or technical explanations that can amuse you for a moment before you forget all about them. Keep an eye open for the following icons throughout the book:



This icon highlights useful information that can save you time (as long as you remember it, of course).



This icon reminds you to do something or emphasizes an important point that you don't want to forget.



Watch out! This icon tells you how to avoid potential headaches and trouble.



This icon identifies the name of a particular file on the CD that contains a sample program printed in the book. By loading the sample program off the CD, you don't have to type the program yourself.



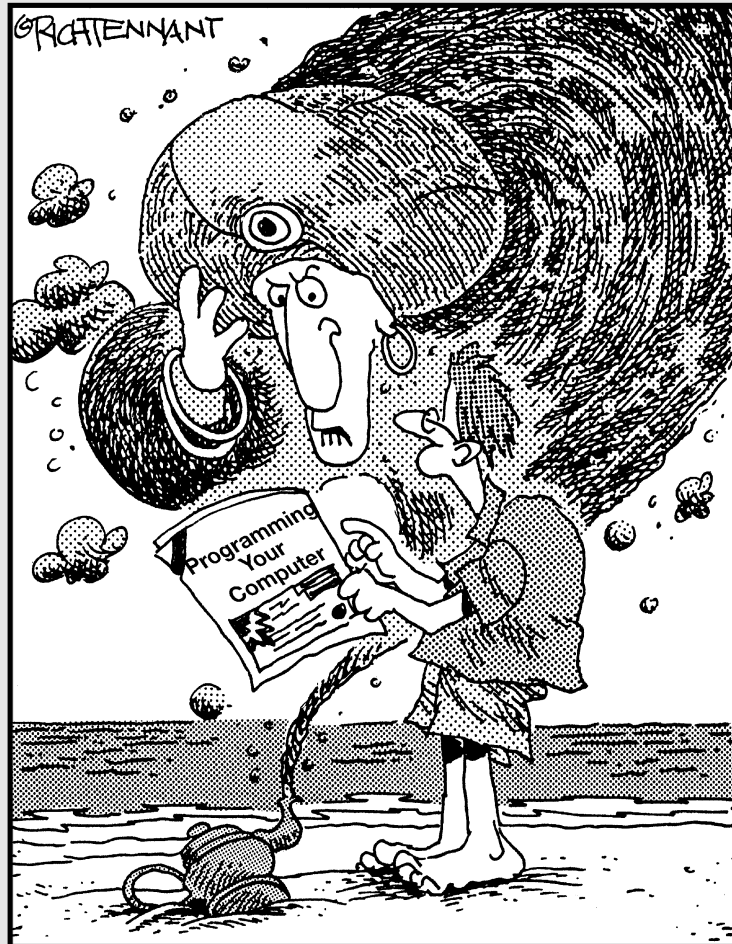
This icon highlights information that's nice to know but which you can safely ignore if you choose. (If you want to become a real programmer, however, you need to cram your brain with as much technical information as possible so that you can fit in with the rest of the programmers in the world.)

Part I

Programming a Computer

The 5th Wave

By Rich Tennant



"Can't I just give you riches or something?"

In this part . . .

Figuring out how to program a computer may seem intimidating, so this part of the book gently guides you through the wonderful world of computer programming. First, you see exactly what programs do and how professionals write programs.

Next, you discover why so many different programming languages exist and why some are more popular than others. You get to know the different tools that programmers use to create, edit, and distribute a program from start to finish.

Finally, this part shows you what to consider if you decide to write a program. You see the pros and cons of using different programming languages. You also find out how people can write programs even if they possess very little programming experience.

By the time that you finish this part of the book, you'll have a better idea of how to write a program, what steps to follow, and how to convert your idea for a program into an actual working product that you can sell or give away for others to use. Who knows? With a little bit of imagination and a lot of persistence, you may create the next program that makes so much money that you can start your own software company and make a million bucks.

Chapter 1

Learning Computer Programming for the First Time

In This Chapter

- ▶ Learning computer programming
 - ▶ Understanding how a computer program works
 - ▶ Knowing how to program a computer
-

Despite what you may have heard, programming a computer isn't difficult. Computer programming is a skill that anyone can pick up, given enough practice, patience, and caffeinated beverages.

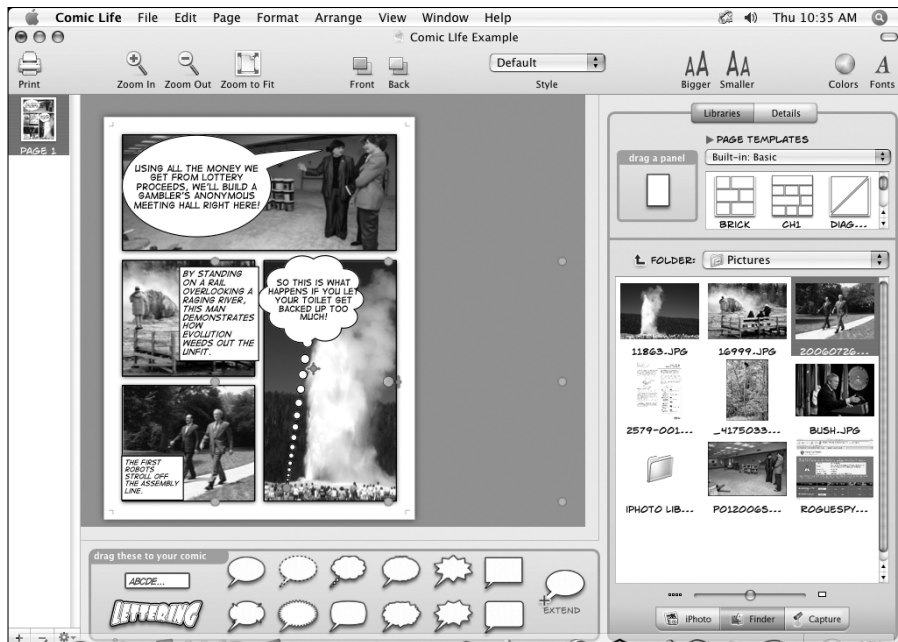
Although computers may seem like tremendously complex electronic beasts, relax. Few people know how internal-combustion engines work, yet people can still figure out how to drive a car. Similarly, anyone can pick up programming skills without worrying (too much) about the specific details that make a computer work.

Why Learn Computer Programming?

The first question that you (or your friends, co-workers, and relatives) may ask is, "Why bother learning to program a computer?" The answer depends on your ultimate goals, but the following list offers some common answers to consider:

- ✓ **For fun:** People learn skiing, dancing, gardening, scuba diving, and painting because they enjoy the experience. They may never become professionals or experts in their chosen hobbies, but they enjoy fiddling around nevertheless. Similarly, programming a computer can prove fun because you might design a simple program that displays your boss's ugly face on the computer. More complex programs may make you a million dollars so that you never again need to work for a boss with an ugly face. Figure 1-1 shows a program known as *Comic Life*, which can turn any photograph into a comic book. This program was written in a programming language called Objective-C on a Macintosh.

Figure 1-1:
Comic Life is
a program
that can
turn any
picture into
a comic
book
cartoon.



- ✓ **To fill a need:** Many people learn programming with no intention of becoming full-time, professional programmers. They just want a program that solves a particular problem, but they can't find a program that does it, so they write the program themselves. A man once needed a program to help him file his taxes, for example, but he couldn't find one, so he taught himself programming and wound up creating TurboTax, one of the most popular tax-preparation programs in the country. Similarly, a freelance writer got frustrated with trying to write a novel with an ordinary word processor, so he created a word processor, specially designed to organize a story, called Z-Write, as shown in Figure 1-2. Z-Write was written by using a program called REALbasic. Whatever your interests, you can write a program to solve a specific problem that others may find useful as well.
- ✓ **For a new or second career:** With computers taking over the world, you're never unemployed for long if you know how to program a computer. Companies are always looking to create new programs, but you also find a growing market for programmers who can maintain and modify the millions of existing programs that do everything from storing hotel reservations to transferring bank deposits electronically. If you know how to program a computer, you're in a much better position to