

Making Everything Easier!™

Objective-C® Programming

FOR DUMMIES®

Learn to:

- Create apps for iPhone®, iPad®, and OS X® using object-oriented programming
- Use Xcode® 4 for debugging and code completion
- Identify and eliminate errors in your programs
- Structure applications and use Objective-C syntax

Neal Goldstein
Karl Kowalski



```
NSDictionary* englandDictionary = [app  
[NSFileManager defaultManager] fileExistsAtPath: balancePath])  
if ([[NSFileManager defaultManager  
ray alloc] initWithContentsOfFile: balancePath];  
ry valueForKey:@"Europe"];
```

Get More and Do More at Dummies.com®



Start with **FREE** Cheat Sheets

Cheat Sheets include

- Checklists
- Charts
- Common Instructions
- And Other Good Stuff!

To access the Cheat Sheet created specifically for this book, go to
www.dummies.com/cheatsheet/objectiveprogramming

Get Smart at Dummies.com

Dummies.com makes your life easier with 1,000s of answers on everything from removing wallpaper to using the latest version of Windows.

Check out our

- Videos
- Illustrated Articles
- Step-by-Step Instructions

Plus, each month you can win valuable prizes by entering our Dummies.com sweepstakes. *

Want a weekly dose of Dummies? Sign up for Newsletters on

- Digital Photography
- Microsoft Windows & Office
- Personal Finance & Investing
- Health & Wellness
- Computing, iPods & Cell Phones
- eBay
- Internet
- Food, Home & Garden

Find out “HOW” at Dummies.com

*Sweepstakes not currently available in all countries; visit Dummies.com for official rules.



Objective-C[®]
Programming
FOR
DUMMIES[®]

Objective-C[®] ***Programming*** FOR ***DUMMIES[®]***

by Neal Goldstein and Karl G. Kowalski



WILEY

John Wiley & Sons, Inc.

Objective-C® Programming For Dummies®

Published courtesy of EMC Corporation by
John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2013 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published by John Wiley & Sons, Inc., Hoboken, New Jersey

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 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. Objective-C is a registered trademark of Apple, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, 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 publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2012949144

ISBN 978-1-118-21398-8 (pbk); ISBN 978-1-118-22878-4 (ebk); ISBN 978-1-118-26609-0 (ebk); ISBN 978-1-118-23128-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1



About the Authors

Neal Goldstein is a recognized leader in making state-of-the-art and cutting-edge technologies practical for commercial and enterprise development. He was one of the first technologists to work with commercial developers at firms such as Apple Computer, Lucasfilm, and Microsoft to develop commercial applications using object-based programming technologies. He was a pioneer in moving that approach into the corporate world for developers at Liberty Mutual Insurance, USWest (now Verizon), National Car Rental, EDS, and Continental Airlines, showing them how object-oriented programming could solve enterprise-wide problems. His book (with Jeff Alger) on object-oriented development, *Developing Object-Oriented Software for the Macintosh* (Addison Wesley, 1992), introduced the idea of scenarios and patterns to developers. He was an early advocate of the Microsoft .NET framework, and he successfully introduced it into many enterprises, including Charles Schwab. He was one of the earliest developers of Service Oriented Architecture (SOA), and as Senior Vice President of Advanced Technology and the Chief Architect at Charles Schwab, he built an integrated SOA solution that spanned the enterprise, from desktop PCs to servers to complex network mainframes. (He holds four patents as a result.) As one of IBM's largest customers, he introduced the folks at IBM to SOA at the enterprise level and encouraged them to head in that direction.

Since the release of the iPhone SDK in March 2008, he has been focusing on mobile applications. He has had eight applications in the App Store. These include a series of Travel Photo Guides (developed with his partners at mobilefortytwo), and a Digital Field Guides series (<http://lp.wiley.com/DestinationDFGiPhoneApp>), developed in partnership with John Wiley & Sons. He also has a free app called Expense Diary that allows you to keep track of things like expenses, mileage, and time by adding them to your calendar.

He has developed mobile strategies for a number of businesses, ranging from National Cinemedia to the American Automobile Association (AAA). His strategies focus on Mobile 2.0 — integrating mobile across the enterprise, creating a consistent user experience across devices and applications in an application ecosystem, and developing a user experience architecture that both leverages — and is constrained by — the device. He has spent the last three years working with mobile device users and developers to determine what makes mobile devices so appealing, what users want from an application on a phone or tablet, and what makes an app compelling. These efforts have resulted in the Application Ecosystem model for mobile applications and an underlying Model Application Controller Architecture based on web services that has become a key element in his client work and his books.

In his copious spare time, he also teaches introductory and advanced classes on iPhone and iPad development (for clients as well as some public classes) and does expert witness work.

Along with those apps and his consulting, he has written several books on iPhone programming, *iPhone Application Development For Dummies* (multiple editions) (Wiley), *Objective-C For Dummies* (Wiley), and he co-authored (with Tony Bove) *iPad Application Development For Dummies* (including multiple editions) (Wiley) and *iPhone Application Development All-in-One For Dummies* (Wiley). He's also the primary author (with Jon Manning and Paris Buttfield-Addison) of *iPhone & iPad Game Development For Dummies*.

Karl Kowalski has traveled the world of computers and software development for far longer than he's willing to admit. He has written programs for airplanes, robots, games, and even particle accelerators, and he has developed software on platforms ranging from desktop computers to mainframes all the way down to smartphones. He is also the author of *Mac Application Development For Dummies* (John Wiley & Sons, Inc., 2011). He lives near Boston and works for RSA, The Security Division of EMC, where he develops security solutions for mobile platforms such as iPhone and BlackBerry, and desktop operating systems such as Windows and Mac OS X. In his spare time, he develops software for smartphones as part of a startup, BlazingApps. And if there are any spare seconds in the day, he does some voice-over work for one of his favorite journals, *The Objective Standard*.

Dedication

To my friends and family, and especially my children Sarah and Evan and my wife, Linda. She deserves special recognition for her support and patience and for maintaining her (and my) sense of humor. Without her support I never would have been able to write 12 books in 3.5 years. Thank you so much.

This is for you
As day follows night
The tunnel ends in the light
Finally time to play

— *Neal Goldstein*

To my incredible family: Connie, Stanley, Lee Anne, David, Rosemarie, Joseph, Candi, Reeseling the Nieceling, and Mason — thank you for all your love and support.

— *Karl Kowalski*

Author's Acknowledgments

Acquisitions Editor Kyle Looper for extending the vision of this book and keeping this project on track. Thanks to Project Editor Charlotte Kughen for guiding us through the process and keeping everything on track and Copy Editor John Edwards for helping us make things clear. Tech Editor Jesse Feiler kept us on our toes, and his comments made this a better book. Great job, team, and I deeply appreciate all of your work.

Thanks again to my agent Carole Jelen for her continued work and support in putting and keeping these projects together.

— *Neal Goldstein*

Publisher's Acknowledgments

We're proud of this book; please send us your comments at <http://dummies.custhelp.com>. For other comments, 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.

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

Acquisitions and Editorial

Project Editor: Charlotte Kughen,
The Wordsmithery LLC
Senior Acquisitions Editor: KyleLooper
Copy Editor: John Edwards
Technical Editor: Jesse Feiler
Editorial Manager: Jodi Jensen
Editorial Assistant: Leslie Saxman
Sr. Editorial Assistant: Cherie Case
Cover Photo: © John Shepherd/iStockphoto.com
Cartoons: Rich Tennant
(www.the5thwave.com)

Composition Services

Project Coordinator: Katherine Crocker
Layout and Graphics: Joyce Haughey,
Julie Trippetti
Proofreaders: Lauren Mandelbaum,
Bonnie Mikkelson
Indexer: BIM Indexing & Proofreading Services

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

Kathleen Nebenhaus, Vice President and Executive Publisher

Composition Services

Debbie Stailey, Director of Composition Services

Contents at a Glance

<i>Introduction</i>	<i>1</i>
<i>Part I: Getting to the Starting Line</i>	<i>7</i>
Chapter 1: Computer Programming Exposed!.....	9
Chapter 2: Creating Your First Program	21
Chapter 3: The Object in Objective-C.....	63
<i>Part II: Speaking the Language of Objective-C.....</i>	<i>71</i>
Chapter 4: Language and Programming Basics	73
Chapter 5: Functions and Data Structures	111
Chapter 6: Adding a Little More Class to Your Program	145
Chapter 7: Objects Objects Everywhere.....	175
Chapter 8: Using the Debugger	193
Chapter 9: Using Control Statements and Loops.....	211
<i>Part III: Walking the Object-Oriented Walk.....</i>	<i>233</i>
Chapter 10: Basic Inheritance	235
Chapter 11: Encapsulating Objects	255
Chapter 12: The Birth of an Object.....	275
Chapter 13: Getting a Handle on Memory Management.....	289
<i>Part IV: Moving from Language to Application.....</i>	<i>307</i>
Chapter 14: Getting Data from Other Objects.....	309
Chapter 15: Show Me the Data	323
Chapter 16: Extending the Behavior of Objects.....	357
Chapter 17: Adding an iPhone User Interface.....	373
Chapter 18: Adding a Mac User Interface.....	403
<i>Part V: The Part of Tens</i>	<i>419</i>
Chapter 19: Ten Debugging Tips.....	421
Chapter 20: Ten Ways to Be a Happy Developer	425
<i>Index</i>	<i>431</i>

Table of Contents

***Introduction* 1**

About This Book	2
Conventions Used in This Book	2
Foolish Assumptions	3
How This Book Is Organized	3
Part I: Getting to the Starting Line	3
Part II: Speaking the Language of Objective-C	4
Part III: Walking the Object-Oriented Walk	4
Part IV: Moving from Language to Application	4
Part V: The Part of Tens	5
Icons Used in This Book	5
On the Web	6
Where to Go from Here	6

***Part I: Getting to the Starting Line* 7**

Chapter 1: Computer Programming Exposed! 9

Why a Computer Program Is Like a Recipe	10
Examining a simple computer program	12
Understanding How Computer Languages Work	14
Creating a computer program	14
Running a computer program	15
What Is Objective-C, Anyway?	17
Understanding programming languages	18
Running your program in a runtime environment	19
Using frameworks and libraries	19
Your suite of development tools	20
Using Xcode 4.4	20
Using Objective-C Version 2.0	20

Chapter 2: Creating Your First Program 21

Getting Started with the Software Development Kit	21
Downloading Xcode	22
Creating Your Xcode Project	26
Exploring your project	31
Workspace areas	32
Displaying an area's content	35
The toolbar and Tab bar	37

The First Program Project	39
The Project editor	39
The Project navigator	41
Building and Running Your Application	44
Building an app	46
All that stuff in main.m	47
Customizing Xcode to Your Liking	50
Getting to Know the Xcode Text Editor	53
Accessing Documentation	54
Quick Help	54
The header file for a symbol	55
Accessing Quick Help	55
Help menu	56
Find	58
On the Web	60
Working with the Examples	61

Chapter 3: The Object in Objective-C 63

Not All Modules Are Created Equal	64
Understanding How Objects Behave	65
Seeing the Concepts in Action	67
Encapsulation	67
Polymorphism	69
Reusable Code	70

Part II: Speaking the Language of Objective-C 71

Chapter 4: Language and Programming Basics 73

It All Comes Down to Your Statements	73
Syntax	75
Semantics	75
Understanding How Variables Work	75
Data types	76
Identifier	78
Initialization	79
Giving Instructions with Operators	80
Really learning how operators (and everything else)	
work in Objective-C	80
Using arithmetic operators	81
Back to variables — floats	89
Bitwise operators	92
Compound assignment operators	93
Increment and decrement operators	95
Comma operator	96

Cast operator.....	97
sizeof operator	97
Making Logical Decisions in Your Code	99
Boolean types	99
Relational and equality operators	103
Logical operators	103
Conditional operator	104
Accessing Data with Pointers	105
Using Constants	106
Declared constants (const)	107
Defined constants (#define)	108
Knowing the Objective-C Reserved Words	109
Congratulations	110
Chapter 5: Functions and Data Structures	111
Thinking about an Application	111
Defining and Declaring Data Structures.....	116
Using Defined Data Types.....	119
Writing Functions	122
Scope of variables.....	130
Unions	132
Enumerations (enum)	133
Declaring Function Prototypes	134
Extending the Functionality of a Program.....	136
Thinking about Extensibility and Enhanceability.....	142
Chapter 6: Adding a Little More Class to Your Program	145
Grasping Objects and Their Classes	145
Moving from Functions and Global Data to Objects and Classes	146
Creating the Interface	148
Declaring the class interface	149
Scoping instance variables	155
The Implementation — Coding the Methods.....	156
Adding the implementation compiler directive.....	156
Defining the createBudget: method.....	157
Defining the rest of the methods	158
Exploring the Program Logic	159
Coding the functionality in the main function	160
Building and running the application.....	164
Extending the program	164
Spreading the Wealth across Files	165
Knowing the Naming Conventions	172
Using id and nil	173

Chapter 7: Objects Objects Everywhere175

Replacing Numbers with Objects	176
Revisiting the documentation	178
Using an NSNumber as an argument	180
Taking Advantage of Array Objects	181
Container classes	182
Tiptoeing through an array	184
Adding mutable arrays	186
Working with fixed arrays	189
Using C Arrays	190
Expanding to multidimensional arrays	191
Finishing up with the main function	192

Chapter 8: Using the Debugger193

Identifying the Usual Suspects	193
Catching syntax errors	194
Fixing it with Live Issues and Fix-It	196
Crashing with runtime errors	197
Dealing with logic errors	198
Using the Debugger	200
Using Breakpoints	204
Using the Static Analyzer	207

Chapter 9: Using Control Statements and Loops211

Creating Transaction Objects	212
Managing all those objects	213
Adding the Transaction class	213
Using switch Statements	216
Using Loop Statements to Repeat Instructions	220
The for loop	221
The while loop	223
The do while loop	225
Adding Loops to Your Program	226
Building the New Application	229
Taking the Leap: Jump Statements	230
Knowing the Problems with switch Statements	231

Part III: Walking the Object-Oriented Walk 233**Chapter 10: Basic Inheritance235**

Replacing a Control Structure with Polymorphism	236
How inheritance works	238
Knowing what inheritance enables you to do	240
Using inheritance effectively	241

Implementing Inheritance in a Program	242
Creating the Transaction superclass	242
Adding the files for the new subclasses	245
Implementing the new subclasses	246
Modifying main to use the new classes	248
Considering Polymorphism and Inheritance	252
Chapter 11: Encapsulating Objects	255
Getting to Know the Model-View-Controller (MVC) Pattern	255
Implementing the MVC Pattern	257
Get out of main	258
Get into main	259
Creating a New Project	260
Creating the Destination Class	263
Designing the destination	264
Implementing the methods	265
Modifying the Budget class	267
Removing UI type functionality from the Transaction objects	269
Coding the New main	271
Yes, Another Two Steps Forward and One Step Back	274
Chapter 12: The Birth of an Object	275
Allocating Objects	275
Initializing Objects	276
Invoking the superclass's init method	278
Initializing instance variables	280
Returning self	281
The Designated_INITIALIZER	287
Chapter 13: Getting a Handle on Memory Management	289
Raising and Terminating Responsible Objects	290
Understanding the object life cycle	291
Understanding Reference Counting	292
Managing memory, retro-style	292
Using ARC	296
Rule 1: Do not send retain, release, or autorelease messages to your objects	297
Rule 2: Do not store pointers to Objective-C objects in C structures (structs)	298
Rule 3: Inform the compiler about ownership when using Core Foundation-type objects	299
Rule 4: Use the @autoreleasepool keyword	299
Rule 5: Follow the naming conventions	299
Converting Non-ARC Code to ARC	300

Part IV: Moving from Language to Application 307**Chapter 14: Getting Data from Other Objects309**

Getting Data from Objects	309
Working with Declared Properties	311
Adding properties	311
Implementing declared properties	312
Accessing the instance variables from within the class	314
Using Accessors to Get Data from Objects	316
Properly Using Properties	320

Chapter 15: Show Me the Data323

Understanding Application-Based Data.....	323
Defining property lists.....	324
Working with property lists.....	325
Using Dictionaries	326
Understanding a dictionary's keys and values	326
Creating a dictionary	327
Adding a plist to Your Project	329
Using plists	332
Creating a mutable dictionary.....	335
Creating, initializing, and using the symbol string	336
Dictionaries of Dictionaries.....	337
Creating a more complex plist	337
Managing a dictionary of dictionaries.....	343
Modifying the plist.....	345
Adding a new entry to the plist.....	345
Updating the dictionary	347
Saving Data in a Separate File	352
Saving Objects as Objects	356

Chapter 16: Extending the Behavior of Objects357

Understanding Delegation.....	358
Using Protocols.....	359
Declaring a protocol.....	359
Creating a protocol.....	360
Adding delegation to Transaction	361
Adopting a protocol.....	364
Categories.....	366
Using categories.....	371
Defining informal protocols.....	371

Chapter 17: Adding an iPhone User Interface 373

Creating Your Project	374
Using Interface Builder to Create a User Interface.....	376
Implementing the User Interface	385
Adding outlets	390
Connecting Targets with Actions.....	393
Adding the methods	397
Running iVacation in the Simulator	400
A Final Note	401

Chapter 18: Adding a Mac User Interface 403

Creating Your Project	403
Using Interface Builder to Create a User Interface.....	406
Implementing the User Interface	409
Using the Assistant Editor to add outlets and Targets-Actions...	411
Implementing the methods.....	415
Running mVacation on the Mac.....	416
Knowing What's Left to Do	417
The End of the Beginning	417

Part V: The Part of Tens* 419*Chapter 19: Ten Debugging Tips 421**

Check for Semicolons.....	421
“Right” Is Not Always “right”	422
When You’ve Blown It, You’ve Blown It	422
Compiler Warnings Are for Your Own Good.....	422
Get Friendly with Your Debugger	422
Messages to nil.....	423
Dialing a Wrong Number.....	423
Create a “Paper” Trail	423
Incrementally Test.....	424
Use Your Brain.....	424

Chapter 20: Ten Ways to Be a Happy Developer 425

Keep Everyone in the Dark.....	425
Make Your Code Easy to Understand	426
Remember Memory	426
Start by Initializing the Right Way	427
Take Advantage of the Documentation	427

Code, Code, Code	427
Understand That Development Is Not Linear	428
Do It Right from the Start If You Need to Do It Right from the Start	428
Avoid the Code Slinger Mentality	429
The Software Isn't Finished Until the Last User Is Dead.....	429
Keep It Fun.....	430
 <i>Index</i>	 431

Introduction

When the folks at John Wiley & Sons approached me about writing *Objective-C Programming For Dummies*, I thought long and hard about it. Within 480 pages, I wanted to be sure that I could explain to someone with no programming experience how to actually create useful programs.

So I started to think about what makes programming so difficult.

It isn't the concept of how programs work, which I cover easily in Part I. And it isn't really the language itself (or the instruction set — I cover that in Chapter 4). It isn't even the user interface — all that code needed to open and close windows, process menus and the mouse and user touches, draw graphics, and play audio and video (did I leave anything out?). No, while all that used to be really hard, now it's made much easier by using the frameworks available with Mac OS X and iOS.

What is really hard, after you understand the language and framework, is how you structure your program — how you actually go about taking your idea for an application and turning it into a robust Objective-C application.

Finding out how to use the tools is (relatively) easy; knowing how to use them to create a useful application is the real challenge.

So, besides explaining the instruction set and everything else involved with coding, what I do along the way is explain the other things you need to know (things like application architecture and design) — those things that will make it possible for you, when you are done with this book, to go out and start developing your first application. Nothing less.

So instead of a book that only shows you *how* to use all the features (instructions and frameworks) available to you, I decided to write a book that shows you both *how* and *why*. I do that by having you start to develop an application in Chapter 5 (after I go over the instruction set) and add to that same application until you end up with it running on both the iPhone and Mac in Chapters 17 and 18. Granted, this application isn't the most exciting one in the world, but it gives you the opportunity to use every feature of Objective-C that you'll need to know to go out and build your own killer app. What's more, you build the application incrementally, just as a professional develops a commercial application. Occasionally, you will enter some code only to delete it later, which may seem annoying at times. However, you will get a flavor for how you'll work when you are out on your own.

And while some development will be annoying and tedious, in general it is fun. So go enjoy yourself while you're finding out about Objective-C. I know I do.

About This Book

Objective-C Programming For Dummies is a beginner's guide to developing applications for both iOS devices and the Mac. You don't need any programming experience to get started. I expect you to come as a blank slate, ready to be filled with useful information and new ways to do things. In some ways, the less you know, the easier it will be for you because you won't have any preconceived notions about programming.

This book distills the hundreds (or even thousands) of pages of Apple documentation, not to mention my own development experience, into only what's necessary to start you developing real applications. I explain not only the language, but also along the way, I explicitly talk about object-oriented principles and how doing things in a certain way (that is, following those principles) leads to more extensible and enhanceable programs, which you will discover is the holy grail of programming.

Conventions Used in This Book

This book guides you through the process of building applications by using Objective-C.

Code examples in this book appear in a monospaced font so that they stand out a bit better. This means that the code you see will look like this:

```
NSLog(@"I am an Objective-C statement.");
```

Objective-C is based on C, which (I want to remind you) *is* case sensitive, so please enter the code that appears in this book *exactly* as it appears in the text. I also use the standard Objective-C naming conventions — for example, class names always start with a capital letter, and the names of methods and instance variables always start with a lowercase letter.

All URLs in this book appear in a monospaced font as well:

```
www.dummies.com
```

If you're ever uncertain about anything in the code, you can always look at the source code at the website associated with this book: www.dummies.com/go/objectprogrammingfd. And from time to time, I provide updates for the code and post other things you might find useful on my website, www.nealgoldstein.com.

Foolish Assumptions

To find out how to program in Objective-C for Mac OS X or iOS, you need a Macintosh computer with the latest version of the Mac OS on it. You also need to download the Software Development Kit (SDK). You don't have to become a registered Apple Developer to do that, but registering with Apple gives you access to all the information Apple provides for understanding Objective-C, iOS, and Mac OS X. I show you how to do both in Chapter 2 (don't worry; it doesn't cost a cent).

I assume that you don't have any programming knowledge but that you have at least a passing acquaintance with some of the ideas, and more important, a desire to know how to program. In general, the code is easy and straightforward (the book isn't written to dazzle you with fancy coding techniques).

I also assume that you're familiar with the Mac and/or iPhone and that you are comfortable doing all the things you have to do on the Mac to run applications, including using the Finder to cruise the file system to see what's there.

How This Book Is Organized

Objective-C Programming For Dummies has five main parts.

Part I: Getting to the Starting Line

Part I introduces you to the world of application development. You find out how programs work and what you have to do to take an idea and turn it into a computer program. I explain the tools that are available to you and describe how to use them, and I lead you through downloading the Software Development Kit (SDK), which includes Xcode (Apple's development environment for the OS X and iOS operating systems). You get up and running on your first application, which gives you a taste for what words like *compiling* and *building* mean. You also find out how to become a registered Apple Developer, both for the Mac and the iPhone (and if you are an iPhone developer, what you are required to do to distribute your applications through Apple's App Store).

Part II: Speaking the Language of Objective-C

As with any other skill, you have to pay your dues, and that means understanding the instruction set of the language and knowing how to use some of the languagelike features made available to you in the frameworks. You start by building an application that you will add to as you discover more and more about Objective-C.

Think of this as getting down the vocabulary of a new language, but without the pain and all that memorization.

Part III: Walking the Object-Oriented Walk

After you understand the basic instruction set and the other Objective-C and framework features, it's time to put those instructions together to create a program. In this part, I focus on the right way to structure your program — what's known as the *program architecture*. Having the right architecture results in a program that not only works but also can be extended to add new functionality easily. And not only that, it enables you to easily track down and fix those pesky bugs that make their home in everyone's programs. I also show you how to deal with the mundane, but necessary, plumbing issues such as memory management and object initialization.

Whereas Part II is about getting down the vocabulary, Part III is about using the vocabulary to create sentences and paragraphs and even entire books.

Part IV: Moving from Language to Application

With an architecture in place, you can begin to add more and more functionality to your program. You start to work with data and discover some of the tricks that framework developers use to make their frameworks so extensible.

After you have your application doing what you want it to do, you need to take all that functionality and make it available to the user. So, in this part, I show you how your application fits into the user SDK-supplied frameworks that do all the user interface heavy lifting on Mac OS X and iOS. And because you design the application the right way from the start, you'll be able to plug it into the user interface with minimal effort. You just do some building of the

user interface in Interface Builder (part of the SDK), add a few lines of code, and you are there. No sweat, no bother. And yes, because you did it the right way from the start, the same application code will run on both the Mac and iOS devices (using the frameworks for Mac OS X and iOS).

Part V: The Part of Tens

Part V consists of voices from the trenches. I also show you some tips on debugging (yes, your application will, upon occasion, have bugs) that might shorten those late, into-the-night debugging sessions that are (unfortunately) part and parcel of being a developer. Although they may not always be fun, solved bugs are often a great source of conversation among developers. I also offer some tips about approaching application development that will lead to good health and happiness as a developer.

Icons Used in This Book



When you see this icon, you can be sure that the code available at this book's website applies to the current example. You can find the code for all projects in this book at www.dummies.com/go/objcprogrammingfd — perfect for those who don't feel like typing the code.



This icon indicates a useful pointer that you shouldn't skip.



This icon represents a friendly reminder. It describes a vital point that you should keep in mind while proceeding through a particular section of the chapter.



This icon signifies that the accompanying explanation might be informative (dare I say, interesting), but it isn't essential to understanding Objective-C application development. Feel free to skip past these tidbits if you like (though skipping while trying to absorb the main concepts may be tricky).



This icon alerts you to potential problems that you may encounter along the way. Read and obey these bits of experience to avoid trouble.

On the Web

I've uploaded all the code for the projects in this book to the website at www.dummies.com/go/objcprogrammingfd. You'll find folders for each chapter starting with Chapter 4. In each of these chapter folders is a folder that contains the Xcode project that provides the starting point for that chapter. So for Chapter 4, for example, the folder is labeled `Chapter 4 Start Here`.

Also in the chapter folder is a folder that contains the final version of the project for each chapter (except for Chapter 4 where it isn't applicable). Some chapters also have intermediate versions of the project; so, in Chapter 5, for example, the folders are `Chapter 5 A`, `Chapter 5 B`, and so on. I explain what is in each in the corresponding chapter.

Where to Go from Here

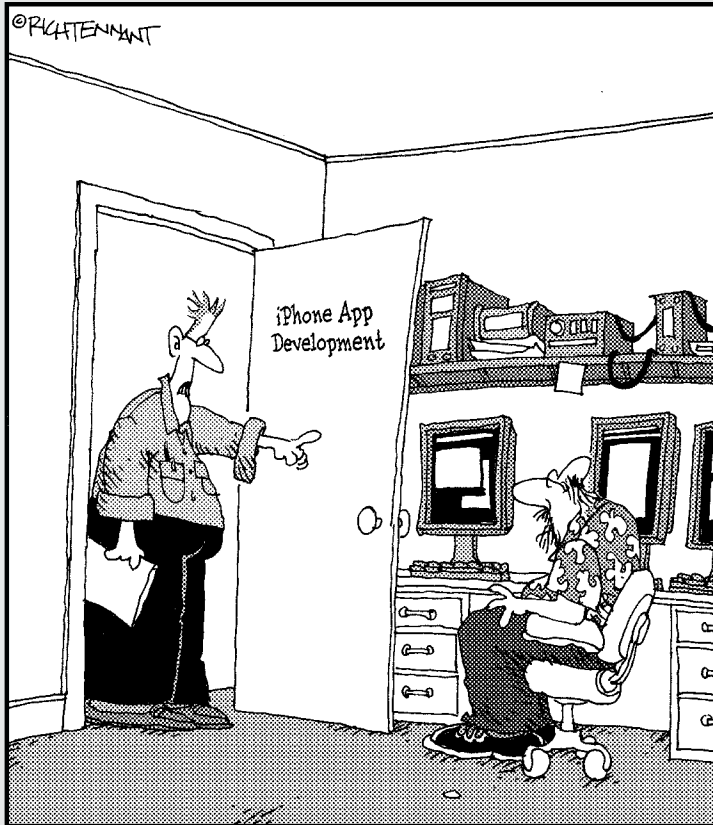
Occasionally, we have updates to our technology books. If this book does have technical updates, they will be posted at www.dummies.com/go/objcprogrammingfdupdates.

Part I

Getting to the Starting Line

The 5th Wave

By Rich Tennant



"Stop working on the Priority Parking Spot Allocation program. They want to fast track the Coffee Pot/Cubicle Proximity program."

In this part . . .

So you've decided you want to learn to program. You may have a good idea for a Mac or iPhone application and realize that the first thing you need to do is find out how to program in Objective-C. And although you may have a vague idea about it, you know you're going to have to learn exactly what programming is and what's required to create an application.

In this part, I help you understand what you need to know to get started. First of all, how do applications even work? How do you translate your ideas into a computer language that tells the computer what you want it to do, and then how does it take those instructions and actually do them? What is all this compiler and framework stuff, and what exactly is object-oriented programming?

You find out what makes a good application and what you can do to make yours a good one. Finally, so you can get free development software from Apple, I take you through the process of registering as an Apple developer. I explain how you can download the Software Development Kit (SDK), and even how to build your first program.