

*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**



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



# Objective-C<sup>®</sup> Programming For Dummies<sup>®</sup>

Visit

[www.dummies.com/cheatsheet/objectiveprogramming](http://www.dummies.com/cheatsheet/objectiveprogramming) to view this book's cheat sheet.

## Table of Contents

[Introduction](#)

[About This Book](#)

[Conventions Used in This Book](#)

[Foolish Assumptions](#)

[How This Book Is Organized](#)

[Part I: Getting to the Starting Line](#)

[Part II: Speaking the Language of Objective-C](#)

[Part III: Walking the Object-Oriented Walk](#)

[Part IV: Moving from Language to Application](#)

[Part V: The Part of Tens](#)

[Icons Used in This Book](#)

[On the Web](#)

[Where to Go from Here](#)

[Part I: Getting to the Starting Line](#)

## [Chapter 1: Computer Programming Exposed!](#)

### [Why a Computer Program Is Like a Recipe](#)

#### [Examining a simple computer program](#)

### [Understanding How Computer Languages Work](#)

#### [Creating a computer program](#)

#### [Running a computer program](#)

### [What Is Objective-C, Anyway?](#)

#### [Understanding programming languages](#)

#### [Running your program in a runtime environment](#)

#### [Using frameworks and libraries](#)

#### [Your suite of development tools](#)

#### [Using Xcode 4.4](#)

#### [Using Objective-C Version 2.0](#)

## [Chapter 2: Creating Your First Program](#)

### [Getting Started with the Software Development Kit](#)

#### [Downloading Xcode](#)

### [Creating Your Xcode Project](#)

#### [Exploring your project](#)

#### [Workspace areas](#)

[Displaying an area's content](#)

[The toolbar and Tab bar](#)

[The First Program Project](#)

[The Project editor](#)

[The Project navigator](#)

## [Building and Running Your Application](#)

[Building an app](#)

[All that stuff in main.m](#)

[Customizing Xcode to Your Liking](#)

[Getting to Know the Xcode Text Editor](#)

[Accessing Documentation](#)

[Quick Help](#)

[The header file for a symbol](#)

[Accessing Quick Help](#)

[Help menu](#)

[Find](#)

[On the Web](#)

[Working with the Examples](#)

## [Chapter 3: The Object in Objective-C](#)

[Not All Modules Are Created Equal](#)

[Understanding How Objects Behave](#)

[Seeing the Concepts in Action](#)

[Encapsulation](#)

[Polymorphism](#)



## Reusable Code

## Part II: Speaking the Language of Objective-C

### Chapter 4: Language and Programming Basics

#### It All Comes Down to Your Statements

Syntax

Semantics

#### Understanding How Variables Work

Data types

Identifier

Initialization

#### Giving Instructions with Operators

Really learning how operators (and everything else) work in Objective-C

Using arithmetic operators

Back to variables — floats

Bitwise operators

Compound assignment operators

Increment and decrement operators

Comma operator

Cast operator

sizeof operator

#### Making Logical Decisions in Your Code

Boolean types

[Relational and equality operators](#)  
[Logical operators](#)  
[Conditional operator](#)

[Accessing Data with Pointers](#)  
[Using Constants](#)

[Declared constants \(const\)](#)  
[Defined constants \(#define\)](#)

[Knowing the Objective-C Reserved Words](#)  
[Congratulations](#)

## [Chapter 5: Functions and Data Structures](#)

[Thinking about an Application](#)  
[Defining and Declaring Data Structures](#)  
[Using Defined Data Types](#)  
[Writing Functions](#)

[Scope of variables](#)  
[Unions](#)  
[Enumerations \(enum\)](#)

[Declaring Function Prototypes](#)  
[Extending the Functionality of a Program](#)  
[Thinking about Extensibility and](#)  
[Enhanceability](#)

## [Chapter 6: Adding a Little More Class to Your Program](#)

[Grasping Objects and Their Classes](#)

[Moving from Functions and Global Data to Objects and Classes](#)  
[Creating the Interface](#)

[Declaring the class interface](#)  
[Scoping instance variables](#)

[The Implementation — Coding the Methods](#)

[Adding the implementation compiler directive](#)  
[Defining the createBudget: method](#)  
[Defining the rest of the methods](#)

[Exploring the Program Logic](#)

[Coding the functionality in the main function](#)  
[Building and running the application](#)  
[Extending the program](#)

[Spreading the Wealth across Files](#)  
[Knowing the Naming Conventions](#)  
[Using id and nil](#)

[Chapter 7: Objects Objects Everywhere](#)

[Replacing Numbers with Objects](#)

[Revisiting the documentation](#)  
[Using an NSNumber as an argument](#)

[Taking Advantage of Array Objects](#)

[Container classes](#)  
[Tiptoeing through an array](#)  
[Adding mutable arrays](#)  
[Working with fixed arrays](#)

## [Using C Arrays](#)

[Expanding to multidimensional arrays](#)  
[Finishing up with the main function](#)

## [Chapter 8: Using the Debugger](#)

### [Identifying the Usual Suspects](#)

[Catching syntax errors](#)  
[Fixing it with Live Issues and Fix-It](#)  
[Crashing with runtime errors](#)  
[Dealing with logic errors](#)

[Using the Debugger](#)  
[Using Breakpoints](#)  
[Using the Static Analyzer](#)

## [Chapter 9: Using Control Statements and Loops](#)

### [Creating Transaction Objects](#)

[Managing all those objects](#)  
[Adding the Transaction class](#)

[Using switch Statements](#)  
[Using Loop Statements to Repeat Instructions](#)

[The for loop](#)  
[The while loop](#)  
[The do while loop](#)

[Adding Loops to Your Program](#)  
[Building the New Application](#)  
[Taking the Leap: Jump Statements](#)  
[Knowing the Problems with switch Statements](#)

## [Part III: Walking the Object-Oriented Walk](#)

### [Chapter 10: Basic Inheritance](#)

[Replacing a Control Structure with Polymorphism](#)

[How inheritance works](#)  
[Knowing what inheritance enables you to do](#)  
[Using inheritance effectively](#)

### [Implementing Inheritance in a Program](#)

[Creating the Transaction superclass](#)  
[Adding the files for the new subclasses](#)  
[Implementing the new subclasses](#)  
[Modifying main to use the new classes](#)

### [Considering Polymorphism and Inheritance](#)

### [Chapter 11: Encapsulating Objects](#)

[Getting to Know the Model-View-Controller \(MVC\) Pattern](#)  
[Implementing the MVC Pattern](#)

[Get out of main](#)  
[Get into main](#)

[Creating a New Project](#)  
[Creating the Destination Class](#)

[Designing the destination](#)  
[Implementing the methods](#)  
[Modifying the Budget class](#)  
[Removing UI type functionality from the Transaction objects](#)

[Coding the New main](#)  
[Yes, Another Two Steps Forward and One Step Back](#)

[Chapter 12: The Birth of an Object](#)

[Allocating Objects](#)  
[Initializing Objects](#)

[Invoking the superclass's init method](#)  
[Initializing instance variables](#)  
[Returning self](#)

[The Designated\\_INITIALIZER](#)

[Chapter 13: Getting a Handle on Memory Management](#)



## Raising and Terminating Responsible Objects

### Understanding the object life cycle

## Understanding Reference Counting

### Managing memory, retro-style

## Using ARC

Rule 1: Do not send retain, release, or autorelease messages to your objects

Rule 2: Do not store pointers to Objective-C objects in C structures (structs).

Rule 3: Inform the compiler about ownership when using Core Foundation-type objects

Rule 4: Use the @autoreleasepool keyword

Rule 5: Follow the naming conventions

## Converting Non-ARC Code to ARC

## Part IV: Moving from Language to Application

### Chapter 14: Getting Data from Other Objects

#### Getting Data from Objects

#### Working with Declared Properties

##### Adding properties

##### Implementing declared properties

[Accessing the instance variables from within the class](#)

[Using Accessors to Get Data from Objects](#)  
[Properly Using Properties](#)

## [Chapter 15: Show Me the Data](#)

[Understanding Application-Based Data](#)

[Defining property lists](#)  
[Working with property lists](#)

[Using Dictionaries](#)

[Understanding a dictionary's keys and values](#)  
[Creating a dictionary](#)

[Adding a plist to Your Project](#)

[Using plists](#)  
[Creating a mutable dictionary](#)  
[Creating, initializing, and using the symbol string](#)

[Dictionaries of Dictionaries](#)

[Creating a more complex plist](#)  
[Managing a dictionary of dictionaries](#)

[Modifying the plist](#)

[Adding a new entry to the plist](#)

[Updating the dictionary](#)

[Saving Data in a Separate File](#)

[Saving Objects as Objects](#)

## [Chapter 16: Extending the Behavior of Objects](#)

[Understanding Delegation](#)

[Using Protocols](#)

[Declaring a protocol](#)

[Creating a protocol](#)

[Adding delegation to Transaction](#)

[Adopting a protocol](#)

[Categories](#)

[Using categories](#)

[Defining informal protocols](#)

## [Chapter 17: Adding an iPhone User Interface](#)

[Creating Your Project](#)

[Using Interface Builder to Create a User Interface](#)

[Implementing the User Interface](#)

[Adding outlets](#)

[Connecting Targets with Actions](#)

[Adding the methods](#)

[Running iVacation in the Simulator](#)

[A Final Note](#)

## Chapter 18: Adding a Mac User Interface

Creating Your Project

Using Interface Builder to Create a User Interface

Implementing the User Interface

Using the Assistant Editor to add outlets and Targets-Actions

Implementing the methods

Running mVacation on the Mac

Knowing What's Left to Do

The End of the Beginning

## Part V: The Part of Tens

### Chapter 19: Ten Debugging Tips

Check for Semicolons

“Right” Is Not Always “right”

When You've Blown It, You've Blown It

Compiler Warnings Are for Your Own Good

Get Friendly with Your Debugger

Messages to nil

Dialing a Wrong Number

Create a “Paper” Trail

Incrementally Test

Use Your Brain

### Chapter 20: Ten Ways to Be a Happy Developer

Keep Everyone in the Dark

Make Your Code Easy to Understand  
Remember Memory  
Start by Initializing the Right Way  
Take Advantage of the Documentation  
Code, Code, Code  
Understand That Development Is Not Linear  
Do It Right from the Start If You Need to Do It  
Right from the Start  
Avoid the Code Slinger Mentality  
The Software Isn't Finished Until the Last User  
Is Dead  
Keep It Fun

Cheat Sheet

***Objective-C® Programming  
For Dummies®***

**by Neal Goldstein and Karl  
G. Kowalski**



John Wiley & Sons, Inc.



## **Objective-C® Programming For Dummies®**

Published courtesy of EMC Corporation

**John Wiley & Sons, Inc.**

111 River St.

Hoboken, NJ 07030-5774

[www.wiley.com](http://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](http://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](http://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.wileypub.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:** Kyle Looper

**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](http://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

# 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:

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/objcprogrammingfd](http://www.dummies.com/go/objcprogrammingfd). 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](http://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.