Objective-C for Absolute Beginners

iPhone, iPad, and Mac Programming Made Easy
Second Editon

Gary Bennett
Mitch Fisher
Brad Lees

Apress®

Objective-C for Absolute Beginners: iPhone, iPad, and Mac Programming Made Easy, Second Edition

Copyright © 2011 by Gary Bennett, Mitch Fisher, Brad Lees

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-3653-5

ISBN-13 (electronic): 978-1-4302-3654-2

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning
Lead Editors: Michelle Lowman and Matthew Moodie
Technical Reviewer: James Bucanek
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Engel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Kelly Moritz
Copy Editor: Scribendi, Inc.
Compositor: MacPS, LLC
Indexer: BIM Indexing & Proofreading Services
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com` or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/bulk-sales`.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Any source code or other supplementary materials referenced by the author in this text is available to readers at `www.apress.com`. For detailed information about how to locate your book’s source code, go to `http://www.apress.com/source-code/`.

I'd like to dedicate this book to the two people who had the biggest impact on my career: my dad, Don W. Bennett, and Steve Jobs. Both passed away this year. Thanks for inspiring me to work in the field where I can have fun, make a difference, be creative, and live the American Dream.

—Gary Bennett

I would like to thank my family and friends who have always supported my endeavors. I would especially like to thank Heather, Matthew, and my two children Eric and Jade for patiently living without me for the many nights and forever-busy weekends. I would also like to thank my friends Gary and Brad for all the help they provided. It's been great working with them again.

—Mitch Fisher

I would like to thank my wife Natalie and my kids for the support and time they have given me to work on this book. I am also grateful for good friends who persuade me to take on crazy endeavors.

—Brad Lees

Contents at a Glance

Contents v
About the Authors x
About the Technical Reviewer xi
Acknowledgments xii
Introduction xiii
Chapter 1: Becoming a Great iOS or Mac Programmer 1
Chapter 2: Programming Basics 13
Chapter 3: It's All About the Data 39
Chapter 4: Making Decisions About…and Planning Program Flow 63
Chapter 5: Object Oriented Programming with Objective-C 87
Chapter 6: Learning Objective-C and Xcode 103
Chapter 7: Objective-C Classes, Objects, and Methods 129
Chapter 8: Programming Basics in Objective-C 163
Chapter 9: Comparing Data 199
Chapter 10: Creating User Interfaces 215
Chapter 11: Storing Information 237
Chapter 12: Protocols and Delegates 261
Chapter 13: Memory, Addresses, and Pointers 267
Chapter 14: Introducing the Xcode Debugger 291
Index 309

Contents

Contents at a Glance iv
About the Authors x
About the Technical Reviewer xi
Acknowledgments xii
Introduction xiii

Chapter 1: Becoming a Great iOS or Mac Programmer 1
Thinking like a Developer 1
Completing the Development Cycle 5
Introducing Object Oriented Programming 6
Working with the Alice Interface 9
Summary 11
Exercises 12

Chapter 2: Programming Basics 13
Taking a Tour with Alice 13
Navigation Menu 14
World Window 15
Classes, Objects, and Instances in Alice 18
Object Tree 18
Editor Area 19
Details Area 20
Events Area 20
Creating an Alice App—To the Moon Alice 21
Your First Objective-C Program 27
Launching and Using Xcode 4.2 27
Summary 36
Exercises 37

Chapter 3: It's All About the Data 39
Numbering Systems Used in Programming 39
Bits 39
Bytes 41
Hexadecimal 43

Unicode 44
Data Types 44
Using Variable and Data Types with Alice 45
Data Types and Objective-C 54
Identifying Problems 60
Summary 62
Exercises 62

Chapter 4: Making Decisions About…and Planning Program Flow 63

Boolean Logic 63
Truth Tables 65
Comparison Operators 67
Designing Apps 68
Pseudo-Code 68
Design Requirements 70
Flowcharting 73
Designing and Flowcharting an Example App 74
The App's Design 75
Using Loops to Repeat Program Statements 76
Coding the Example App in Alice 78
Coding the Example App in Objective-C 79
Nested If Statements and Else-If Statements 83
Removing Extra Characters 83
Improving the Code Through Refactoring 83
Running the App 84
Moving Forward Without Alice 84
Summary 85
Exercises 86

Chapter 5: Object Oriented Programming with Objective-C 87

The Object 88
What Is a Class? 89
Planning Classes 90
Planning Properties 90
Planning Methods 92
Implementing the Classes 94
Inheritance 99
Why Use OOP? 100
It is everywhere 100
Eliminate Redundant Code 100
Ease of Debugging 101
Ease of Replacement 101
Advanced Topics 101
Interface 101
Polymorphism 101
Summary 102
Exercises 102

Chapter 6: Learning Objective-C and Xcode 103

A Brief History of Objective-C 103

Understanding the Language Symbols 104
Putting the "Objective" into Objective-C 105
Writing Another Program in Xcode 111
Creating the Project 111
Summary 127
Exercises 128

Chapter 7: Objective-C Classes, Objects, and Methods 129

Creating an Objective-C Class 129
Declaring Interfaces and Instance Variables 131
Sending Messages (Methods) 131
Working with the Implementation File 134
Coding Your Methods 136
Using Your New Class 138
Creating Your Project 138
Adding Objects 141
Writing the Implementation File 144
Creating the User Interface 147
Hooking Up the Code 152
Running the Program 157
Taking Class Methods to the Next Level 158
Accessing the Xcode Documentation 159
Summary 160
Exercises 160

Chapter 8: Programming Basics in Objective-C 163

Collections 163
Using NSSet 164
Using NSArray 165
NSDictionary 166
Determining Class Type in a Collection 167
Using the Mutable Classes 168
NSMutableSet 168
NSMutableArray 169
NSMutableDictionary 170
Creating the BookStore Application 171
Introducing Instance Variables 176
Accessing Instance Variables 177
Using Getter and Setter Methods 178
Introducing Properties 180
Using Properties 182
Understanding the Importance of Conventions 182
Finishing the MyBookstore Program 183
Creating the View 183
Adding Instance Variables 185
Adding a Description 186
Creating a Simple Data Model Class 189
Modifying the MasterViewController 191
Modifying the DetailViewController 196

Summary 197
Exercises....198
Chapter 9: Comparing Data 199
Revisiting Boolean Logic....199
Using Relational Operators 200
Comparing Numbers....200
Creating an Example Xcode App....202
Using Boolean Expressions....206
Comparing Strings....207
Comparing Dates 209
Combining Comparisons....211
Using the Switch Statement 212
Summary 213
Exercises....214
Chapter 10: Creating User Interfaces 215
Understanding Interface Builder 216
The Model-View-Controller....217
Human Interface Guidelines (HIGs) 219
Creating an Example iPhone App with Interface Builder 220
Using Interface Builder 225
The Dock....225
The Library....226
Inspector Pane and Selector Bar 228
Creating the View....229
Using Outlets....230
Connecting Actions and Objects 232
Implementation File 233
Summary 235
Exercises....235
Chapter 11: Storing Information.... 237
Storage Considerations....237
Preferences....238
Writing Preferences 238
Reading Preferences....239
Databases 240
Storing Information in a Database 240
Getting Started with Core Data 241
The Model 244
Managed Object Context....252
Setting Up the Interface....252
Summary 259
Exercises....260
Chapter 12: Protocols and Delegates 261
Multiple Inheritance 261
Understanding Protocols....262
Protocol Syntax....263
Understanding Delegates....264

Next Steps....265
Summary....265
Chapter 13: Memory, Addresses, and Pointers....267
Understanding Memory....268
Bits, Bytes, and Bases....268
Converting Base-10 (Decimal) to Base-2 (Binary)....269
Using Base-16 (Hexadecimal) Numbering....270
Understanding Memory Address Basics....272
Allocating Memory....276
Working with Automatic Variables and Pointers....276
Deallocating Memory....277
Using Special Pointers....279
Managing Memory in Objective-C with ARC....280
Managing Memory in Objective-C Without ARC....282
Using the Retain/Release Model....283
Working with Implied Retain Messages and Autorelease....285
Sending the dealloc Message....286
If Things Go Wrong....287
A Note About ARC....289
Summary....289
Exercises....289
Chapter 14: Introducing the Xcode Debugger....291
Getting Started with Debugging....292
Setting Breakpoints....293
Using the Breakpoint Navigator....294
Debugging Basics....296
Working with the Debugger Controls....297
Using the Step Controls....298
Looking at the Thread Window and Call Stack....300
Debugging Variables....300
Dealing with Code Errors and Warnings....302
Warnings....304
Summary....306
Index....309

About the Authors

Gary Bennett is president of xcelMe.com. xcelMe.com provides iPhone/iPad programming courses online. Gary has taught thousands of students how to develop iPhone/iPad apps, and has several very popular apps in the iTunes App Store. Gary's students have some of the best-selling apps in the iTunes App Store. Gary also worked for 25 years in the technology and defense industries. He served 10 years in the US Navy as a nuclear engineer aboard two nuclear submarines. After leaving the Navy, Gary worked for several companies as a software developer, CIO, and president. As CIO, he helped take VistaCare public in 2002. Gary also coauthored *iPhone Cool Projects* for Apress. He lives in Scottsdale, Arizona, with his wife Stefanie and their four children.

Mitch Fisher is a software developer in the Phoenix, Arizona, area. He was introduced to PCs back in the 1980s when 640 KB was considered more memory than anyone could ever use. Over the last 25 years, Mitch has worked for several large and medium-sized companies as a software engineer, software architect, and software manager, and has led teams of developers on multimillion-dollar projects. Mitch now divides his time between writing iOS applications, creating server-side UNIX technologies, and teaching iOS development at `xcelMe.com`.

Brad Lees has more than 14 years of experience in application development and server management. He has specialized in creating and initiating software programs in real estate development systems and financial institutions. Highlights of his professional career include his positions as information systems manager at The Lyle Anderson Company, product development manager for Smarsh, vice president of application development for iNation, and information technology manager at The Orcutt/Winslow Partnship, the largest architectural firm in Arizona. A graduate of Arizona State University, Brad resides in Phoenix with his wife Natalie and their five children.

About the Technical Reviewer

James Bucanek has spent the past 30 years programming and developing microcomputer systems. He has experience with a broad range of technologies, from embedded consumer products to industrial robotics. James currently focuses on Macintosh and iPhone software development. When not programming, James indulges in his love of the arts. He earned an associate degree in classical ballet from the Royal Academy of Dance, and occasionally teaches at Adams Ballet Academy.

Acknowledgments

We would like to thank Apress for all their help in making this book possible. Specifically, we would like to thank Kelly Moritz, our coordinating editor, for helping us stay focused and overcoming many obstacles. Without Kelly, this book would not have been possible.

Special thanks to Matthew Moodie, our development editor, for all his suggestions during the editorial review process to help make this a great book. Thanks to Chandra Clarke and Scribendi, Inc., the copy editors who made the book look great.

We would also like to thank the Alice Community and Carnegie Mellon University for developing Alice and making learning object-oriented programming fun and easy!

Introduction

Over the last three years, we've heard the following countless times:

- "I've never programmed before, but I have a great idea for an iPhone/iPad app."
- "Can I really learn to program the iPhone or iPad?"

We always answer, "Yes, but you have to believe you can." Only you are going to tell yourself you can't do it.

For the Newbie

This book assumes you may have never programmed before. The book is also written for someone who may have never programmed before using object-oriented programming (OOP) languages. There are many Objective-C books out there, but all of these books assume you have programmed before and know OOP and computer logic. We wanted to write a book that takes readers from knowing little or nothing about computer programming and logic to being able to program in Objective-C. After all, Objective-C is the native programming language for the iPhone, iPad, and Mac.

Over the last three years, we have taught well over a thousand students at xcelMe.com to be iPhone/iPad (iOS) developers. Many of our students have developed some of the most successful iOS apps in their category in the iTunes App Store. We have incorporated what we have learned in our first two courses, Introduction to Object-oriented Programming and Logic and Objective-C for iPhone/iPad Developers, into this book.

For the More Experienced

Many developers who programmed years ago or programmed in a non-OOP language need a background in OOP and Logic before they dive into Objective-C. This book is for you. We gently walk you through OOP and how it is used in iOS development to help make you a successful iOS developer.

Why Alice: An Innovative 3D Programming Environment

Over the years, universities have struggled with several issues with their computer science departments:

- High male-to-female ratios
- High drop-out rates

- Longer than average time to graduation

One of the biggest challenges to learning OOP languages like Java, C++, or Objective-C is the steep learning curve from the very beginning. In the past, students had to learn the following topics all at once:

- Object-oriented principles
- A complex Integrated Development Environment (IDE), i.e., Xcode, Eclipse, Visual Studio
- The syntax of the programming language
- Programming logic and principles

As a result, Carnegie Mellon University received a grant from the US government and developed Alice. Alice, an innovative 3D programming environment, makes it easy for new developers to create rich graphical applications. Alice is a teaching tool for students learning to program in an OOP environment. The software uses 3D graphics and a drag-and-drop interface to facilitate a more engaging, less frustrating first programming experience.

Alice enables students to focus on learning the principles of OOP without having to focus on learning a complex IDE and Objective-C principles all at once. You get to focus on each topic individually. This helps students feel a real sense of accomplishment as they progress.

As drag-and-drop programming, Alice removes all the complexity of learning an IDE and programming language syntax. You'll see programming is actually fun, and you can develop very cool and sophisticated apps in Alice.

After we introduce the OOP topic and readers feel comfortable with the material, we then move into Xcode, where you get to use your new OOP knowledge in writing Objective-C applications. This way, you can focus on the Objective-C syntax and language without having to learn OOP at the same time.

Learning Objective-C Without Alice

More than a thousand `xcelMe.com` students have used this book to become successful iOS developers. At the end of each course, we ask our students if the Alice sections in the first four sections were useful. More than half of the students thought using Alice at the beginning of the first four chapters to introduce the chapter was critical to their success. However, some of the students didn't feel they needed the Alice examples at the beginning of the first four chapters.

We have laid out the first four chapters of this book with the first part of each chapter introducing the OOP topic with Alice; the remaining part of the chapter introduces the topic using Objective-C. Thus, you can skip the Alice material if you feel comfortable with the topic.

How This Book Is Organized

You'll notice that we are all about successes in this book. We introduce the OOP and Logic concepts in Alice and then move those concepts to Xcode and Objective-C. Many students are visual or learn by doing. We use both techniques. We'll walk you through topics and concepts with visual examples and then take you through step-by-step examples reinforcing the concepts.

We often repeat topics in different chapters to reinforce what you have learned and apply these skills in new ways. This enables new programmers to reapply development skills and feel a sense of accomplishment as they progress. Don't worry if you feel you haven't mastered a topic. Keep moving forward!

The Formula for Success

Learning to program is an interactive process between your program and you. Just like learning to play an instrument, you have to practice. You must work through the examples and exercises in this book. Understanding the concept doesn't mean you know how to apply it and use it.

You will learn a lot from this book. You will learn a lot from working through the exercises in this book. *However, you will really learn when you debug your programs.* Spending time walking through your code and trying to find out why it is not working the way you want is an unparalleled learning process. The downside of debugging is a new developer can find it especially frustrating. If you have never wanted to throw your computer out the window, you will. You will question why you are doing this, and whether you are smart enough to solve the problem. Programming is very humbling, even for the most experienced developer.

Like a musician, the more you practice the better you get. By practicing, we mean programming! You can do some amazing things as a programmer. The world is your oyster. Seeing your app in the iTunes App Store is one of the most satisfying accomplishments. However, there is a price, and that price is time spent coding and learning.

Having taught more than a thousand students to become iOS developers, we have put together a formula for what makes students successful. Here is our formula for success:

- Believe you can do it. You'll be the only one who says you can't do this. So don't tell yourself that.
- Work through all the examples and exercises in this book.
- Code, code, and keeping coding. The more you code, the better you'll get.
- Be patient with yourself. If you were fortunate enough to have been a 4.0 student who can memorize material just by reading it, this will not happen with Objective-C coding. You are going to have to spend time coding.
- You learn by reading this book. You really learn by debugging your code.
- Use the ***free*** xcelMe.com webinars and YouTube videos explained at the end of this chapter.
- Don't give up!

The Development Technology Stack

We will walk you through the process of understanding the development process for your iOS apps and what technology you need. However, briefly looking at all the pieces together is helpful. For a sample iPhone app in a Table View, see Figure 1.

Figure 1. *The iPhone/iPad technology stack*

Required Software, Materials, and Equipment

One of the great things about Alice is it available on the three main operating systems used today:

- Windows
- Mac
- Linux

The other great thing about Alice is it is free! You can download Alice at `www.Alice.org`.

Operating System and IDE

Although you can use Alice on many platforms, the Integrated Development Environment (IDE) that developers use to develop iOS apps is Xcode. You have to use *an Intel-based Mac to use Xcode and submit apps!* Xcode is free and is available in the Mac App Store.

Software Development Kits

You will need to register as an iOS developer. You can do this at `http://developer.apple.com/iphone`.

When you are ready to upload your app to the iTunes App Store, you will need to pay $99/year.

Dual Monitors

We recommend developers have a second monitor connected to their computer. It is great to step through your code and watch your output window and iPad simulator at the same time on dual independent monitors. Apple hardware makes this easy. Just plug your second monitor into the display port of any Intel-based Mac, with the correct Mini DisplayPort adapter of course, and you have two monitors working independently of one another. See Figure 2. Note that dual monitors are not required. You will just have to organize your open windows to fit on your screen if you don't.

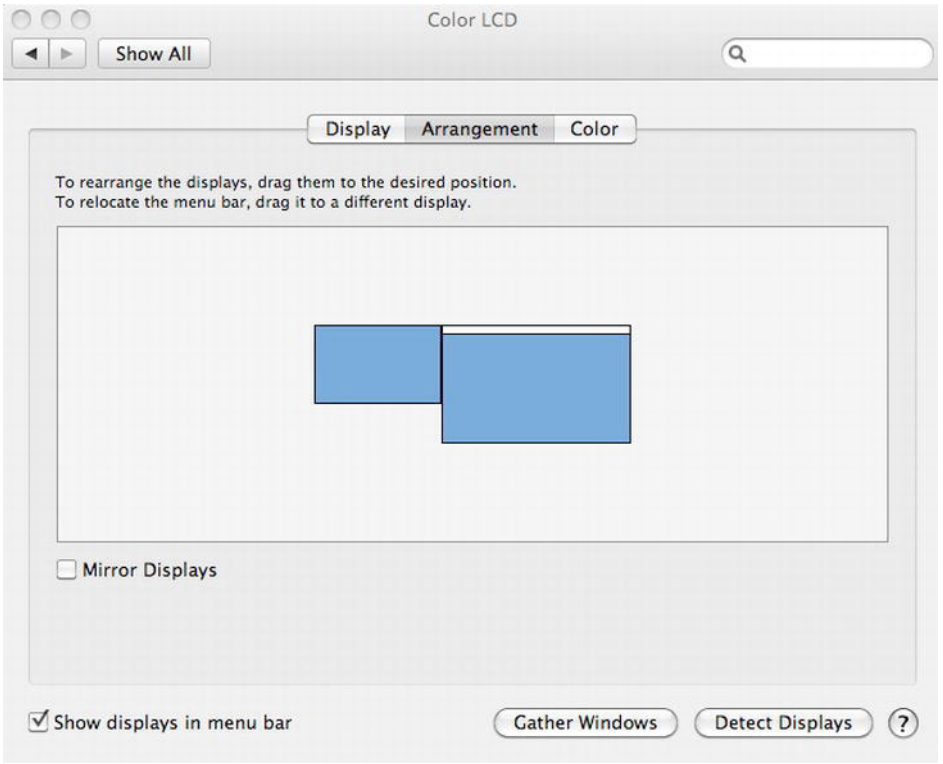


Figure 2. *Dual monitors*

Free Live Webinars, Q&A, and YouTube Videos

Nearly every Wednesday night at 7:30 p.m. Pacific daylight time, we have live webinars and discuss a topic from the book or a timely item of interest. These webinars are free, and you can register for them at `www.xcelme.com/free-webinars.php`.

At the end of the webinars, we do a Q&A. You can ask a question on the topic discussed or any topic in the book.

Additionally, all these webinars are recorded and available on YouTube.

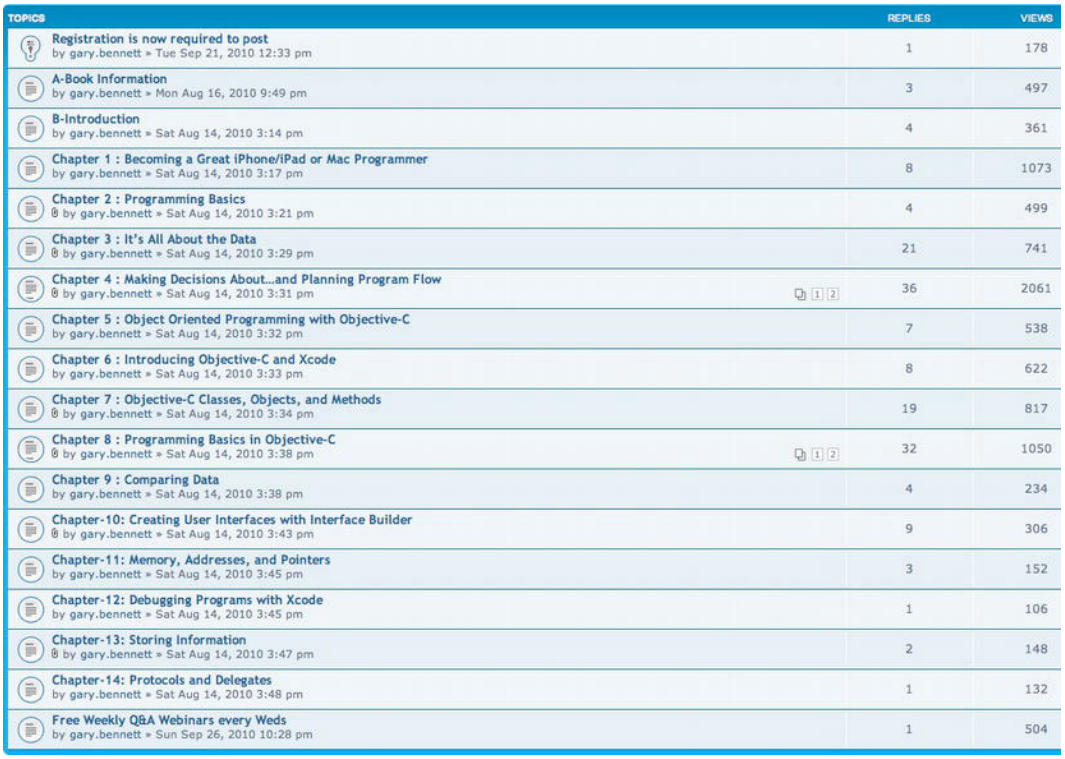
Make sure you subscribe to the YouTube channel so you are notified when new recordings are uploaded.

Figure 3. *Free Objective-C webinars and YouTube videos*

Free Book Forum

We have developed an online forum for this book at `http://forum.xcelme.com`, where you can ask questions while you are learning Objective-C and get answers from the authors. You will also find answers to the exercises and additional exercises to help you learn. See Figure 3.

You can also access answers to exercises and discover helpful links to help you become a successful iPhone/iPad developers and create great apps. See Figure 4. So let's get started!



TOPICS	REPLIES	VIEWS
Registration is now required to post by gary.bennett » Tue Sep 21, 2010 12:33 pm	1	178
A-Book Information by gary.bennett » Mon Aug 16, 2010 9:49 pm	3	497
B-Introduction by gary.bennett » Sat Aug 14, 2010 3:14 pm	4	361
Chapter 1 : Becoming a Great iPhone/iPad or Mac Programmer by gary.bennett » Sat Aug 14, 2010 3:17 pm	8	1073
Chapter 2 : Programming Basics by gary.bennett » Sat Aug 14, 2010 3:21 pm	4	499
Chapter 3 : It's All About the Data by gary.bennett » Sat Aug 14, 2010 3:29 pm	21	741
Chapter 4 : Making Decisions About...and Planning Program Flow by gary.bennett » Sat Aug 14, 2010 3:31 pm 1 2	36	2061
Chapter 5 : Object Oriented Programming with Objective-C by gary.bennett » Sat Aug 14, 2010 3:32 pm	7	538
Chapter 6 : Introducing Objective-C and Xcode by gary.bennett » Sat Aug 14, 2010 3:33 pm	8	622
Chapter 7 : Objective-C Classes, Objects, and Methods by gary.bennett » Sat Aug 14, 2010 3:34 pm	19	817
Chapter 8 : Programming Basics in Objective-C by gary.bennett » Sat Aug 14, 2010 3:38 pm 1 2	32	1050
Chapter 9 : Comparing Data by gary.bennett » Sat Aug 14, 2010 3:38 pm	4	234
Chapter-10: Creating User Interfaces with Interface Builder by gary.bennett » Sat Aug 14, 2010 3:43 pm	9	306
Chapter-11: Memory, Addresses, and Pointers by gary.bennett » Sat Aug 14, 2010 3:45 pm	3	152
Chapter-12: Debugging Programs with Xcode by gary.bennett » Sat Aug 14, 2010 3:45 pm	1	106
Chapter-13: Storing Information by gary.bennett » Sat Aug 14, 2010 3:47 pm	2	148
Chapter-14: Protocols and Delegates by gary.bennett » Sat Aug 14, 2010 3:48 pm	1	132
Free Weekly Q&A Webinars every Weds by gary.bennett » Sun Sep 26, 2010 10:28 pm	1	504

Figure 4. *Reader Forum for accessing answers to exercise and posting questions for authors*