
Objective-C Fundamentals



Christopher K. Fairbairn
Johannes Fahrenkrug
Collin Ruffenach

 MANNING

Objective-C Fundamentals

Objective-C Fundamentals

CHRISTOPHER K. FAIRBAIRN
JOHANNES FAHRENKRUG
COLLIN RUFFENACH



MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact


Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964
Email: orders@manning.com

©2012 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

☉ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964

Development editor: Troy Mott
Technical editor: Amos Bannister
Copyeditor: Linda Kern
Proofreader: Katie Tennant
Typesetter: Dennis Dalinnik
Cover designer: Marija Tudor

ISBN: 9781935182535

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – MAL – 17 16 15 14 13 12 11

brief contents

PART 1	GETTING STARTED WITH OBJECTIVE-C.....	1
1	■ Building your first iOS application	3
2	■ Data types, variables, and constants	28
3	■ An introduction to objects	55
4	■ Storing data in collections	74
PART 2	BUILDING YOUR OWN OBJECTS	95
5	■ Creating classes	97
6	■ Extending classes	124
7	■ Protocols	144
8	■ Dynamic typing and runtime type information	163
9	■ Memory management	177
PART 3	MAKING MAXIMUM USE OF FRAMEWORK FUNCTIONALITY	201
10	■ Error and exception handling	203
11	■ Key-Value Coding and NSPredicate	212
12	■ Reading and writing application data	228
13	■ Blocks and Grand Central Dispatch	257
14	■ Debugging techniques	276

contents

preface xv
acknowledgments xvi
about this book xviii
author online xxi
about the cover illustration xxii

PART 1 GETTING STARTED WITH OBJECTIVE-C1

- 1** ***Building your first iOS application*** 3
- 1.1 **Introducing the iOS development tools** 4
 - Adapting the Cocoa frameworks for mobile devices* 4
 - 1.2 **Adjusting your expectations** 5
 - A survey of hardware specifications, circa mid-2011* 6
 - Expecting an unreliable internet connection* 7
 - 1.3 **Using Xcode to develop a simple Coin Toss game** 7
 - Introducing Xcode—Apple’s IDE* 8
 - Launching Xcode easily* 8 ■ *Creating the project* 9
 - Writing the source code* 12
 - 1.4 **Hooking up the user interface** 15
 - Adding controls to a view* 15 ■ *Connecting controls to source code* 17

- 1.5 Compiling the Coin Toss game 21
- 1.6 Taking Coin Toss for a test run 21
 - Selecting a destination* 22
 - *Using breakpoints to inspect the state of a running application* 23
 - *Running the CoinToss game in the iPhone simulator* 24
 - Controlling the debugger* 25
- 1.7 Summary 27

2 *Data types, variables, and constants* 28

- 2.1 Introducing the Rental Manager application 29
 - Laying the foundations* 29
- 2.2 The basic data types 32
 - Counting on your fingers—integral numbers* 32
 - Filling in the gaps—floating-point numbers* 35
 - Characters and strings* 37
 - *Boolean truths* 39
- 2.3 Displaying and converting values 40
 - NSLog and Format Specifiers* 40
 - *Type casts and type conversions* 43
- 2.4 Creating your own data types 44
 - Enumerations* 44
 - *Structures* 46
 - *Arrays* 48
 - The importance of descriptive names* 50
- 2.5 Completing Rental Manager v1.0, App Store here we come! 52
- 2.6 Summary 54

3 *An introduction to objects* 55

- 3.1 A whirlwind tour of object-oriented programming concepts 56
 - What's wrong with procedural-based languages such as C?* 56
 - What are objects?* 56
 - *What are classes?* 57
 - Inheritance and polymorphism* 57
- 3.2 The missing data type: id 58
- 3.3 Pointers and the difference between reference and value types 59
 - Memory maps* 59
 - *Obtaining the address of a variable* 59
 - Following a pointer* 60
 - *Comparing the values of pointers* 61

- 3.4 Communicating with objects 62
 - Sending a message to an object* 62
 - *Sending a message to a class* 63
 - *Sending nonexistent messages* 64
 - Sending messages to nil* 65
- 3.5 Strings 66
 - Constructing strings* 66
 - *Extracting characters from strings* 67
 - *Modifying strings* 68
 - Comparing strings* 69
- 3.6 Sample application 69
- 3.7 Summary 72

4 Storing data in collections 74

- 4.1 Arrays 75
 - Constructing an array* 75
 - *Accessing array elements* 76
 - Searching for array elements* 77
 - *Iterating through arrays* 79
 - Adding items to an array* 80
- 4.2 Dictionaries 82
 - Constructing a dictionary* 82
 - *Accessing dictionary entries* 84
 - *Adding key/value pairs* 85
 - Enumerating all keys and values* 86
- 4.3 Boxing 88
 - The NSNumber class* 89
 - *The NSValue class* 90
 - nil vs. NULL vs. NSNull* 90
- 4.4 Making the Rental Manager application data driven 91
- 4.5 Summary 94

PART 2 BUILDING YOUR OWN OBJECTS.....95

5 Creating classes 97

- 5.1 Building custom classes 98
 - Adding a new class to the project* 98
- 5.2 Declaring the interface of a class 99
 - Instance variables (ivars)* 100
 - *Method declarations* 101
 - Fleshing out the header file for the CTrentalProperty class* 105
- 5.3 Providing an implementation for a class 106
 - Defining method implementations* 106
 - *Accessing instance variables* 106
 - *Sending messages to self* 107
 - Fleshing out the method file for the CTrentalProperty class* 108

- 5.4 Declared properties 109
 - @property syntax* 109
 - *Synthesizing property getters and setters* 112
 - *Dot syntax* 113
- 5.5 Creating and destroying objects 115
 - Creating and initializing objects* 115
 - *init is pretty dumb* 116
 - Combining allocation and initialization* 118
 - Destroying objects* 119
- 5.6 Using the class in the Rental Manager application 120
- 5.7 Summary 123

6 *Extending classes* 124

- 6.1 Subclassing 124
 - What is subclassing?* 125
- 6.2 Adding new instance variables 127
- 6.3 Accessing existing instance variables 129
 - Manual getters and setters approach* 130
- 6.4 Overriding methods 131
 - Overriding the description method* 132
- 6.5 Class clusters 134
 - Why use class clusters* 134
 - *Multiple public clusters* 135
- 6.6 Categories 136
 - Extending classes without subclassing* 136
 - Using a category* 136
 - *Considerations when using categories* 138
- 6.7 Subclassing in your demo application 138
 - Creating and subclassing CTLease* 139
 - Creating CTPeriodicLease as a subclass of CTLease* 140
 - Creating CTFixedLease as a subclass of CTLease* 141
- 6.8 Summary 143

7 *Protocols* 144

- 7.1 Defining a protocol 145
- 7.2 Implementing a protocol 146
 - Creating the protocol method callers* 147
 - *Making a class conform to a protocol* 148

- 7.3 Important protocols 150
 - <UITableViewDataSource>* 150
 - *<UITableViewDelegate>* 153
 - <UIActionSheetDelegate>* 157
 - *NSXMLParser* 158
- 7.4 Summary 162

8 *Dynamic typing and runtime type information* 163

- 8.1 Static vs. dynamic typing 164
 - Making assumptions about the runtime type* 164
- 8.2 Dynamic binding 166
- 8.3 How messaging works 166
 - Methods, selectors, and implementations* 167
 - Handling unknown selectors* 169
 - *Sending a message to nil* 170
- 8.4 Runtime type information 171
 - Determining if a message will respond to a message* 171
 - Sending a message generated at runtime* 171
 - Adding new methods to a class at runtime* 173
- 8.5 Practical uses of runtime type introspection 174
- 8.6 Summary 176

9 *Memory management* 177

- 9.1 Object ownership 178
- 9.2 Reference counting 179
 - Releasing an object* 180
 - *Retaining an object* 181
 - Determining the current retain count* 182
- 9.3 Autorelease pools 184
 - What is an autorelease pool?* 185
 - *Adding objects to the autorelease pool* 185
 - *Creating a new autorelease pool* 185
 - *Releasing objects in a pool* 187
 - Why not use an autorelease pool for everything?* 187
- 9.4 Memory zones 190
- 9.5 Rules for object ownership 192
- 9.6 Responding to low-memory warnings 193
 - Implementing the *UIApplicationDelegate* protocol* 193
 - Overriding *didReceiveMemoryWarning** 194
 - *Observing the *UIApplicationDidReceiveMemoryWarningNotification** 197
- 9.7 Summary 199

PART 3 MAKING MAXIMUM USE OF FRAMEWORK FUNCTIONALITY.....201

10 *Error and exception handling* 203

- 10.1 NSError—handling errors the Cocoa way 204
 - Getting NSError to talk* 204
 - *Examining NSError’s userInfo Dictionary* 205
- 10.2 Creating NSError objects 206
 - Introducing RentalManagerAPI* 206
 - *Handling and displaying RentalManagerAPI errors* 209
- 10.3 Exceptions 210
 - Throwing exceptions* 210
 - *Catching exceptions* 211
- 10.4 Summary 211

11 *Key-Value Coding and NSPredicate* 212

- 11.1 Making your objects KVC-compliant 213
 - Accessing properties via KVC* 214
 - *Constructing key paths* 215
 - Returning multiple values* 215
 - *Aggregating and collating values* 216
- 11.2 Handling special cases 217
 - Handling unknown keys* 217
 - *Handling nil values* 218
- 11.3 Filtering and matching with predicates 219
 - Evaluating a predicate* 219
 - *Filtering a collection* 220
 - Expressing your predicate condition* 220
 - *More complex conditions* 221
 - *Using key paths in predicate expressions* 222
 - *Parameterizing and templating predicate expressions* 223
- 11.4 Sample application 224
- 11.5 Summary 227

12 *Reading and writing application data* 228

- 12.1 Core Data history 229
 - What does Core Data do?* 229
- 12.2 Core Data objects 231
 - Managed object context* 231
 - *Persistent store coordinator* 231
 - Managed object model* 232
 - *Persistent object store* 232

- 12.3 Core Data resources 232
 - Core Data entities* 232
 - *Core Data attributes* 233
 - Core Data relationships* 234
- 12.4 Building the PocketTasks application 234
 - Examining the Xcode Core Data template* 234
 - Building the data model* 235
 - *Defining the relationships* 236
 - Creating Person entities in pure code* 237
 - *Fetching Person entities in pure code* 239
 - *Adding a master TableView* 240
 - Adding and deleting people* 243
 - *Managing tasks* 246
 - Using model objects* 249
- 12.5 Beyond the basics 251
 - Changing the data model* 251
 - *Performance* 253
 - Error handling and validation* 253
- 12.6 Summary 256

13 *Blocks and Grand Central Dispatch* 257

- 13.1 The syntax of blocks 258
 - Blocks are closures* 260
 - *Blocks and memory management* 262
 - Block-based APIs in Apple's iOS frameworks* 264
- 13.2 Performing work asynchronously 265
 - Meet GCD* 266
 - *GCD fundamentals* 266
 - Building RealEstateViewer* 267
 - *Making the image search asynchronous* 271
 - *Making the image loading asynchronous* 273
- 13.3 Summary 274

14 *Debugging techniques* 276

- 14.1 Building an application, complete with bugs 277
- 14.2 Understanding NSLog 278
- 14.3 Bringing memory leaks under control with Instruments 281
- 14.4 Detecting zombies 284
- 14.5 Summary 286

- appendix A Installing the iOS SDK* 288
- appendix B The basics of C* 293
- appendix C Alternatives to Objective-C* 312
- index* 327

preface

Having been involved in the development of applications on a variety of mobile platforms for more than 10 years, I knew the iPhone was something exciting when it was first introduced back in 2008. From a consumer viewpoint, it had the intangible and hard-to-define elements required to make a compelling device that you just wanted to keep coming back to and interact with. To the user, the device “felt right” and it was a pleasure to use rather than simply being a means to an end to achieve a singular task.

As new and refreshing as the iPhone user experience was, the development tools that supported it were also rather unique. For developers without prior exposure to Apple products, the platform was full of new terms, tools, and concepts to grok. This book is designed to provide an introduction to these technologies, with emphasis on covering only those features available for use by iOS applications. For someone learning a new environment, there’s nothing worse than reading a section of a book and attempting to implement what you learn in an application of your own design, only to realize that the Objective-C or Cocoa feature discussed is only applicable to desktop Mac OS X applications.

I hope you enjoy reading this book and you’ll remember its tips while you develop the next iTunes App Store Top 10 application!

CHRISTOPHER FAIRBAIRN

acknowledgments

A technical book has more than what first meets the eye. A significant number of skills are required to make sure not only that it is technically correct, but that it reads well, looks good, and is approachable by the intended audience. Thus, we thank the entire Manning staff, without whom this book would not exist in its present form. They did more than just correct our errors and polish our words; they also helped make integral decisions about the organization and the contents of the book—decisions that improved it dramatically.

At Manning Publications, we'd like to thank Emily Macel who helped us at an early stage to shape and focus our writing style. Thanks also to Troy Mott, our acquisitions editor, who initially approached us to develop the book and who stayed with us every step of the way. And thanks to Amos Bannister for expertly tech editing the final manuscript during production and for testing the code.

Finally, we'd like to thank the reviewers who generously agreed to read our manuscript as we worked on it; they improved the book immensely: Ted Neward, Jason Jung, Glenn Stokol, Gershon Kagan, Cos DiFazio, Clint Tredway, Christopher Haupt, Berndt Hamboeck, Rob Allen, Peter Scott, Lester Lobo, Frank Jania, Curtis Miller, Chuck Hudson, Carlton Gibson, Emeka Okereke, Pratik Patel, Kunal Mittal, Tyson Maxwell, TVS Murthy, Kevin Butler, David Hanson, Timothy Binkley-Jones, Carlo Bottiglieri, Barry Ezell, Rob Allen, David Bales, Pierre-Antoine Grégoire, Kevin Munc, Christopher Schultz, Carlton Gibson, Jordan Duval-Arnould, Robert McGovern, Carl Douglas, Dave Mateer, Fabrice Dewasmes, David Cuillerier, Dave Verwer, and Glen Marcus.

Christopher would like to thank his fiancée Michele for giving support and encouragement while he worked on this book. She is in many ways an unsung fourth “author” and has contributed greatly. Also, he would like to thank the staff at Manning for their understanding in a trying year involving burglaries, setbacks, and no less than three significant earthquake events. Last but not least, he is thankful for all the support from the important people in his life.

Johannes would like to thank Troy Mott for getting him on board with this project, and Aaron Hillegass for helping him get started with Mac development in the first place, and for being an all-around nice guy. Most of all, he’d like to thank his loving and ever-supportive wife Simone (hey, he already *did* get rid of some of his nerd T-shirts!) and his parents Fred and Petra.

Collin would like to thank Manning Publications for giving him the opportunity to work on this book and the language he is so passionate about. He acknowledges Aaron Hillegass for being a dedicated evangelist for this fantastic language and all its platforms; most of what he knows about Objective-C can be attributed to Aaron’s work. He would like to thank Panic, OmniGraffle, Delicious Library, Rouge Amoeba, MyDreamApp.com, and all the other inspiring software development companies that set such a high bar in the mobile space with their fantastic desktop software. He also thanks ELC Technologies for being so supportive in this endeavor. Thanks to his parents Debbie and Steve for all of their support, and his brothers Brett and Stephen for helping hash out ideas for the book. A big thanks goes to his girlfriend Caitlin for helping him stay dedicated and focused. And finally, he would like to thank Brandon Trebitowski, author with Manning Publications, for his dedication to this platform and for educating young developers.

about this book

Objective-C Fundamentals is an introductory book, intended to complement other books focused on iPhone and iPad application development such as *iOS 4 in Action*. While many books have been written on how to develop iOS applications, most focus on the individual APIs and frameworks provided by the device, rather than the unique language, *Objective-C*, which is a cornerstone of Apple's development platform. To truly master the platform, you must have a strong grip on the language, and that is what this book intends to provide. *Objective-C Fundamentals* is a book that focuses on learning Objective-C in the context of iOS application development. No time is spent discussing aspects or elements of the language that are not relevant to iOS. All examples are fully usable on your own iOS-powered device. We encourage you to read this book straight through, from chapter 1 to chapter 14. This process will introduce the platform, discuss how to program for the iPhone and iPad, and walk you through the entire process step by step.

The audience

We've done our best to make this book accessible to everyone who is interested in creating successful iOS applications using the native Objective-C-based development tools.

If you want to learn about iOS programming, you should have some experience with programming in general. It'd be best if you've worked with C or at least one object-oriented language before, but that's not a necessity. If you haven't, you may find the introduction to the C programming language in appendix B helpful, and you should expect to do some research on your own to bolster your general programming

skills. There's no need to be familiar with Objective-C, Cocoa, or Apple programming in general. We'll give you everything you need to become familiar with Apple's unique programming style. You'll probably have a leg-up if you understand object-oriented concepts; but it's not necessary (and again, you'll find an introduction in chapter 3).

Roadmap

Chapter 1 introduces the tools surrounding Objective-C and iOS application development, and covers the creation of a basic game, ready to run on your device.

Chapter 2 kicks things off by highlighting how data is stored and represented within an Objective-C–based application.

Chapter 3 looks at how Objective-C takes small quantities of data and packages them with logic to form reusable components called *classes*.

Chapter 4 shifts the focus by taking a look at some of the classes, provided out of the box by Cocoa Touch, that can be used to store multiple pieces of related data.

Chapter 5 covers how to create your own custom classes and objects. Learning how to create your own classes is an important building block to becoming a productive developer.

Chapter 6 takes a look at how you can build on top of the foundations provided by an existing class to create a more specialized or customized version of a class without needing to rewrite all of its functionality from scratch.

Chapter 7 discusses how classes can be defined to provide specific functionality, without resorting to requiring all classes to inherit from a common base class. This concept is provided with a language construct called a *protocol*.

Chapter 8 looks deeply at some of the aspects of Objective-C that make it unique. The important distinction between message sending and method invocation is discussed and some powerful programming techniques are demonstrated.

Chapter 9 covers how to keep track of memory allocation within an Objective-C application. Since no automatic garbage collector is available, simple rules are discussed which will allow you to expertly craft applications without introducing memory leaks.

Chapter 10 looks at `NSError` and at some real-life use cases for exceptions, which tools will help you deal with errors gracefully.

Chapter 11 covers Key Value Coding (KVC) and `NSPredicate`-based queries, which are a surprisingly flexible way to filter, search and sort data within Cocoa Touch–based applications.

Chapter 12 gets you started with Core Data and teaches you everything you'll need to know to leverage Core Data for all of your most common data persistence needs.

Chapter 13 introduces a language construct called a *block* and demonstrates this by showing how Grand Central Dispatch (GCD) can be used to simplify multithreaded programming, since it takes care of all the complicated heavy lifting for us.

No application is perfect first time around, so chapter 14 rounds out the book with a discussion on debugging techniques that can help resolve unwanted logic errors and memory leaks quickly and efficiently.

The appendixes contain additional information that didn't fit with the flow of the main text. Appendix A outlines how to enroll in the iOS Developer Program and set up your physical iPhone or iPad device in order to run your own applications on them. Appendix B provides a basic overview of the C programming language that Objective-C is a descendant of. This will be ideal for developers with little experience of a C-based language and those that have previously only developed in languages such as Ruby, Python, or Java. Appendix C outlines some of the alternatives you can use to develop iOS applications, and compares their advantages and disadvantages to Objective-C.

Writing this book was truly a collaborative effort. Chris wrote chapters 1 through 5, 8, 9, 11, 14, and appendixes B and C. Johannes contributed chapters 10, 12, and 13, and appendix A; and Collin was responsible for chapters 6 and 7.

Code conventions and downloads

Code examples appear throughout this book. Longer listings appear under clear listing headings, and shorter listings appear between lines of text. All code is set in a monospace font like `this` to differentiate it from the regular font. Class names have also been set in code font; if you want to type it into your computer, you'll be able to clearly make it out.

With the exception of a few cases of abstract code examples, all code snippets began life as working programs. You can download the complete set of programs from www.manning.com/Objective-CFundamentals. You'll find two ZIP files there, one for each of the SDK programs. We encourage you to try the programs as you read; they include additional code that doesn't appear in the book and provide more context. In addition, we feel that seeing a program work can elucidate the code required to create it.

The code snippets in this book include extensive explanations. We often include short annotations beside the code; and sometimes numbered cueballs beside lines of code link the subsequent discussion to the code lines.

Software requirements

An Intel-based Macintosh running OS X 10.6 or higher is required to develop iOS applications. You also need to download the Xcode IDE and iOS SDK. Xcode is available for purchase in the Mac App Store and the iOS SDK is freely downloadable.

However, the best approach to obtaining Xcode and developing iOS applications is to pay a yearly subscription fee for the iOS Developer Program (<http://developer.apple.com/programs/ios/>). This will provide free access to Xcode and iOS SDK downloads as well as enable testing and deployment of applications on real iPhone and iPad devices, and the iTunes App Store.

author online

Purchase of *Objective-C Fundamentals* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the authors and from other users. To access the forum and subscribe to it, point your web browser to www.manning.com/Objective-CFundamentals. This page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialog between individual readers and between readers and the authors can take place. It is not a commitment to any specific amount of participation on the part of the authors, whose contribution to the Author Online forum remains voluntary (and unpaid). We suggest you try asking the authors some challenging questions lest their interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the cover illustration

On the cover of *Objective-C Fundamentals* is “A man from Tinjan, Istria,” a village in the interior of the peninsula of Istria in the Adriatic Sea, off Croatia. The illustration is taken from a reproduction of an album of Croatian traditional costumes from the mid-nineteenth century by Nikola Arsenovic, published by the Ethnographic Museum in Split, Croatia, in 2003. The illustrations were obtained from a helpful librarian at the Ethnographic Museum in Split, itself situated in the Roman core of the medieval center of the town: the ruins of Emperor Diocletian’s retirement palace from around AD 304. The book includes finely colored illustrations of figures from different regions of Croatia, accompanied by descriptions of the costumes and of everyday life.

In this region of Croatia, the traditional costume for men consists of black woolen trousers, jacket, and vest decorated with colorful embroidered trim. The figure on the cover is wearing a lighter version of the costume, designed for hot Croatian summers, consisting of black linen trousers and a short, black linen jacket worn over a white linen shirt. A gray belt and black wide-brimmed hat complete the outfit.

Dress codes and lifestyles have changed over the last 200 years, and the diversity by region, so rich at the time, has faded away. It’s now hard to tell apart the inhabitants of different continents, let alone of different hamlets or towns separated by only a few miles. Perhaps we have traded cultural diversity for a more varied personal life—certainly for a more varied and fast-paced technological life.

Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of regional life of two centuries ago, brought back to life by illustrations from old books and collections like this one.

Part 1

Getting started with Objective-C

Becoming an iOS application developer can require mastering a number of new tools and technologies such as the Xcode IDE and the Objective-C programming language. Although plenty of step-by-step how-to tutorials are available online for developing small example applications, such as a photo viewer or RSS news feed display application and so on, these typically don't provide much in the way of background information to enable you to develop applications of your own design.

In this part of the book, you'll develop a small game as a learning exercise to become familiar with the development tools surrounding the Objective-C language. As you progress through these chapters, you'll discover more of the meaning and purpose behind the individual steps and tasks outlined in developing the game so you can see the theory and purpose behind each step.

Toward the end of this part, you'll reach a stage where you can confidently create a new project within Xcode and describe the purpose of each file and the meaning behind the various code snippets found within them.

- **[download online Conspiracies \(Repairman Jack, Book 3\)](#)**
- [download A Friend of the Earth for free](#)
- [download Sidesaddle \(The Saddle Club, Book 88\) pdf, azw \(kindle\)](#)
- [Lonely Planet Russia \(7th Edition\) for free](#)
- [download Don't Cry, Tai Lake \(Inspector Chen Cao, Book 7\) for free](#)

- <http://www.khoi.dk/?books/Hungry-Capital--The-Financialization-of-Food.pdf>
- <http://fitnessfatale.com/freebooks/A-Friend-of-the-Earth.pdf>
- <http://hasanetmekci.com/ebooks/Sidesaddle--The-Saddle-Club--Book-88-.pdf>
- <http://sidenoter.com/?ebooks/Lonely-Planet-Russia--7th-Edition-.pdf>
- <http://nautickim.es/books/La-pr--sence-de-Dieu-dans-l-histoire.pdf>