A complete course on C programming for the beginner

Learn C on the Mac

For OS X and iOS

SECOND EDITION

David Mark | James Bucanek

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.

friendsof ED™

Apress®

Contents at a Glance

Introduction

Welcome Aboard

Welcome! Chances are that you are reading this because you love the Mac. And not only do you love the Mac, but you also love the idea of learning how to design and develop your very own Mac programs.

You've definitely come to the right place.

This book assumes that you know how to use your Mac. That's it. You don't need to know anything about programming—not one little bit. We'll start off with the basics, and each step we take will be a small one to make sure that you have no problem following along.

This book will focus on the basics of programming. At the same time, you'll learn the essentials of the C programming language.

In Douglas Adam's book The Hitchhiker's Guide to the Galaxy, the answer to "the Ultimate Question of Life, the Universe, and Everything" is determined to be "42." That answer is, of course, wrong; the correct answer is "C."

The C language is the wellspring of software. The nothing-short-of-miraculous revolution in computing and consumer electronics over the past half century has largely been accomplished using C, languages that are direct descendants of C (Objective-C, C++), or languages designed to work like C (Java, C#). Learn C and the programming world is your oyster.

■ **Note** Douglas Adams was a big Macintosh fan.

Once you get through this book, you'll be ready to move on to object-oriented programming and Objective-C—the official programming language of OS X and iOS.

Does this all sound a little overwhelming? Not to worry; in this book, we'll take small steps, so nobody gets lost. You can definitely do this!

Who Is This Book For?

When Dave wrote the very first edition of *Learn C on the Mac* back in 1991, he was writing with college students in mind. After all, college was where he really learned to program. It seems he was way off.

"My first clue that I had underestimated my audience was when I started getting e-mails from fifth graders who were making their way through the book. Fifth graders! And not just one but lots of nine-, ten-, and eleven-year-old kids were digging in and learning to program. Cool! And the best part of all was when these kids started sending me actual shipping products that they created. You can't imagine how proud I was and still am."

Dave was really on to something. Over the years, we've heard from soccer moms, hobbyists, even folks who were using the Mac for the very first time, all of whom made their way through *Learn C on the Mac* and came out the other end, proud, strong, and full of knowledge.

So what do you need to know to get started? Although learning C by just reading a book is possible, you'll get the most out of this book if you run each example program as you encounter it. To do this, you'll need a Mac running OS X (preferably version 10.6.8 or later) and an Internet connection. You'll need the Internet connection to download the free tools Apple has graciously provided for anyone interested in programming the Mac and to download the projects that go along with this book.

Again, if you know nothing about programming, don't worry. The first few chapters of this book will bring you up to speed. If you have some programming experience (or even a lot), you might want to skim the first few chapters, and then dig right into the C fundamentals that start in Chapter 3.

The Lay of the Land

Here's a quick tour of what's to come in this book.

- Chapter 1 shows you how to get the free software tools you'll use throughout this book.
- Chapter 2 explains some of the basics of how computer programs are built.
- Chapter 3 shows you how to embed a series of programming statements into a reusable function, something you can call again and again.
- Chapter 4 adds variables and operators into the mix, bringing the power of mathematical expressions into your programs.
- Chapter 5 teaches you how to watch your program execute, line-by-line, to see that it's doing the right thing, or fix it if it's not.
- Chapter 6 introduces the concept of flow control, using constructs like if, else, do, and while to control the direction your program takes.
- Chapter 7 covers pointers and parameters, two concepts that will add a dramatic new level of power to your programs.
- Chapter 8 moves beyond the simple data types used in the first half of the book, adding the ability to work with more complex numbers along with data types like arrays and text strings.

- Chapter 9 takes a break to show you how to deploy your finished program and use it from the command line.
- Chapter 10 dives even deeper into data and teaches you how to design your own custom data structures.
- Chapter 11 shows you how to save your program's data and read it back in again by introducing the concept of the data file.
- Chapter 12 gives you some techniques for dealing with errors, for when things go wrong.
- Chapter 13 covers a variety of advanced topics—typecasting, unions, recursion, sorting, collections, and much more.
- Finally, Chapter 14 wraps things up and points you to the next step on your journey.

Ready to get started? Let's go!

Chapter 1

Go Get the Tools!

If you want to build a house, you need a solid set of well-crafted tools. Building computer programs is no different. Programming requires a specialized set of *development tools*—basically, programs that make programs.

In the early days of C, you only needed a few, relatively simple tools. As computers have become more sophisticated, so has the universe of development tools. Today, it's not uncommon to employ dozens of programs to create even a "simple" application: editors, compilers, linkers, debuggers, emulators, profilers, analyzers, and more. Add to that list programs that help you find documentation, cross reference your code, record your development history, and, well, it's starting to look like a whole hardware store full of tools!

The good news is that Apple has come to your rescue. Just as Apple has used an elegant user interface to demystify their most sophisticated applications, they've done the same for software developers. (That's you!)

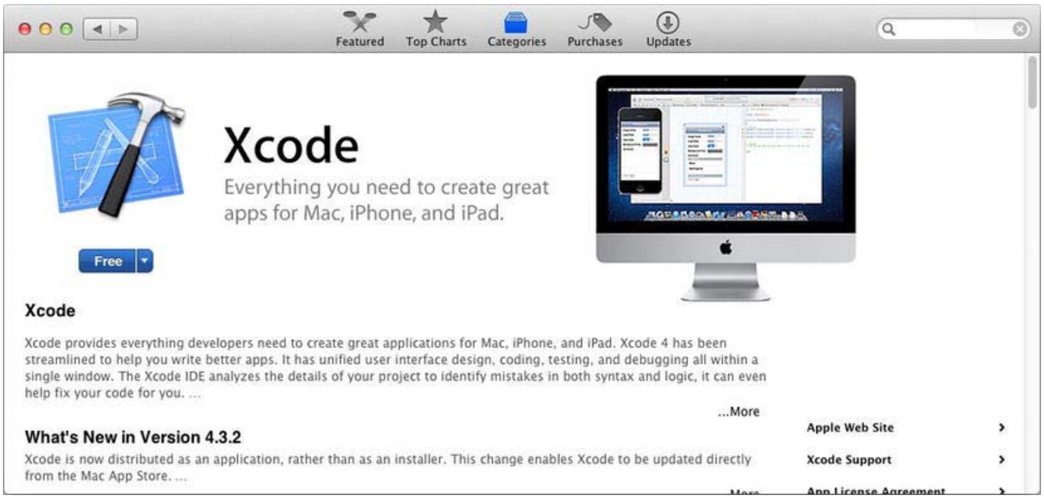
Installing Xcode

Apple's Xcode is a complete hardware store of software development tools, packaged and delivered as a single application. All you have to do is write your program and Xcode will—behind the scenes—direct the scores of individual development tools needed to turn your idea into reality. It would make the Wizard of Oz proud.

NOTE: An application that organizes multiple development tools into a single workspace is called an *integrated development environment* (IDE). Xcode is an IDE.

And getting Xcode into your computer couldn't be easier. The entire Xcode development suite is available from the App Store.

Launch the App Store, go to the Developer Tools category (or just search for "Xcode"), and click to install Xcode, as shown in Figure 1-1. Don't worry if your screen looks a bit different than the figure. Apple is constantly updating Xcode, so there will probably be a new version of Xcode in the App Store by the time this book hits the shelves (or your screen).



Xcode

Everything you need to create great apps for Mac, iPhone, and iPad.

Xcode

Xcode provides everything developers need to create great applications for Mac, iPhone, and iPad. Xcode 4 has been streamlined to help you write better apps. It has unified user interface design, coding, testing, and debugging all within a single window. The Xcode IDE analyzes the details of your project to identify mistakes in both syntax and logic, it can even help fix your code for you. ...

...More

What's New in Version 4.3.2

Xcode is now distributed as an application, rather than as an installer. This change enables Xcode to be updated directly from the Mac App Store. ...

Apple Web Site ›
Xcode Support ›
App License Agreement ›

Figure 1-1. *Installing Xcode from the App Store*

That's it! Sit back and wait for Xcode to download and install. And you're going to have to wait awhile, as it's a really big application. So amuse yourself with the rest of this chapter while it downloads. Switch to the Purchases view, at the top of the App Store window, if you want see how the download is progressing.

How much is that IDE in the Window?

Xcode has gone through various prices in the past. Apple really wants you to create great applications and has strived, for the most part, to make its developments tools freely available.

It used to be that Xcode was only available to *registered developers*. Becoming a registered developer usually costs money, so Xcode was "free" only in the sense that the prize inside a cereal box is "free."

For a while, Xcode was priced at $5. As of this writing, Xcode is free in the App Store. Hopefully, it will stay that way.

NOTE: If you're running an older version of OS X and don't have access to the App Store, you can still download an earlier version of Xcode—but we don't recommend it.

The first problem you're going to encounter is how to get your copy of Xcode. As of this writing, you must be a registered developer to obtain an older version of Xcode. Unfortunately, Apple no longer offers free developer registration—largely because Xcode is now available for free in the App Store—so you'll have to pay to register, and that can be expensive. If you *are* a registered developer or have access to Apple's University Program for higher education, you can log into `http://developer.apple.com/` and download the tools.

But your biggest problem is going to be the differences between the current Xcode and older versions. The code examples in this book will still work and make sense, but the commands, windows, features, and controls are all going to be substantially different. You're going to have to figure out a lot on your own.

We certainly don't want to discourage anyone from learning C on the Mac, but we strongly recommend you upgrade to the latest version of OS X so you have access to the latest version of Xcode.

What's a Registered Developer?

So what's a registered developer and do you need to be one? The short answer is "not yet."

Becoming a registered developer grants you access to even more tools and resources than just Xcode. But you don't need any of that to write great applications for OS X or iOS! You don't need it to use Xcode. You certainly don't need to be a registered developer to work through this book (or most other books, for that matter).

You *will* need to become a registered developer if you want to sell, or even give away, your masterpieces on any of Apple's app stores. How cool would that be? You can register at any time, so there's no hurry. When you are ready, visit `http://developer.apple.com/`.

Getting the Projects

While you're still waiting for Xcode to download and install, why not get the project files for this book? Everything you need to create the projects in this book is described in the text, but downloading the finished projects from the Apress web site will save you a lot of typing.

Go to `http://www.apress.com/book/view/9781430245339`. Below the book's description, you'll see some folder tabs, one of which is labeled `Source Code/Downloads`. Click that tab. Now find the link that downloads the projects for this book. Click that link and a file named `Learn C Projects.zip` will download to your hard drive.

Locate the file `Learn C Projects.zip` in your `Downloads` folder (or wherever the browser saved it). Double-click the file to extract its contents, leaving you with a folder named `Learn C Projects`. Move the folder wherever you like.

Using Xcode

Once Xcode has finished installing, launch it as you would any application, from the dock or LaunchPad. When first launched, Xcode will present its startup window (Figure 1-2).