

Role of an iOS Engineer

by Devmountain Tutorials



WHAT'S COVERED

In this lesson, you will learn how to explore the responsibilities and job characteristics of an iOS engineer. Specifically, this lesson will cover:

Hi, I'm Camilla. I'm the iOS Engineer on the team. I've been here for 11 months. Before Poodle Jumper, I worked at another small tech company in San Francisco. I've been coding iOS apps for a few years now and I have several of my own apps in the iOS App Store. I got interested in coding in college, but I was disappointed they didn't offer more classes for modern programming languages. I was formally trained in the programming language C++. I dabbled in coding between jobs, but I didn't find my groove until a friend showed me an iOS app they made. I didn't know that the programming languages Apple uses were so similar to what I learned. I got involved in Apple's developer community and built my first app in my spare time. It was pretty simple and not a very impressive app, but once it was published, I was hooked. I love how Apple provides an entire ecosystem of tools for developers to use. When I'm not working, I love to tinker with 3D printing. I'm designing my own figures for a tabletop game I'm creating.



1. Pace of Change is Fast

Being an iOS developer is so rewarding! The daily challenges we face evolve because the pace of change in mobile technology is on fire. iOS engineers leverage the same engineering concepts as other programming languages, but there are some differences that I'll explain.

If you want to be an iOS engineer, you need to be willing to be continuously learning. The primary language we write in, Swift, changes every year. So, there are always new frameworks and new syntax to learn. As Swift changes and evolves, so do the size and features of iOS devices. With every new screen size, resolution, or change to the device features, we need to learn how to adjust our views and code to match the new iOS feel.



THINK ABOUT IT

Do you remember when Apple released the iPhone X in 2017? There was a debacle about the "notch" at

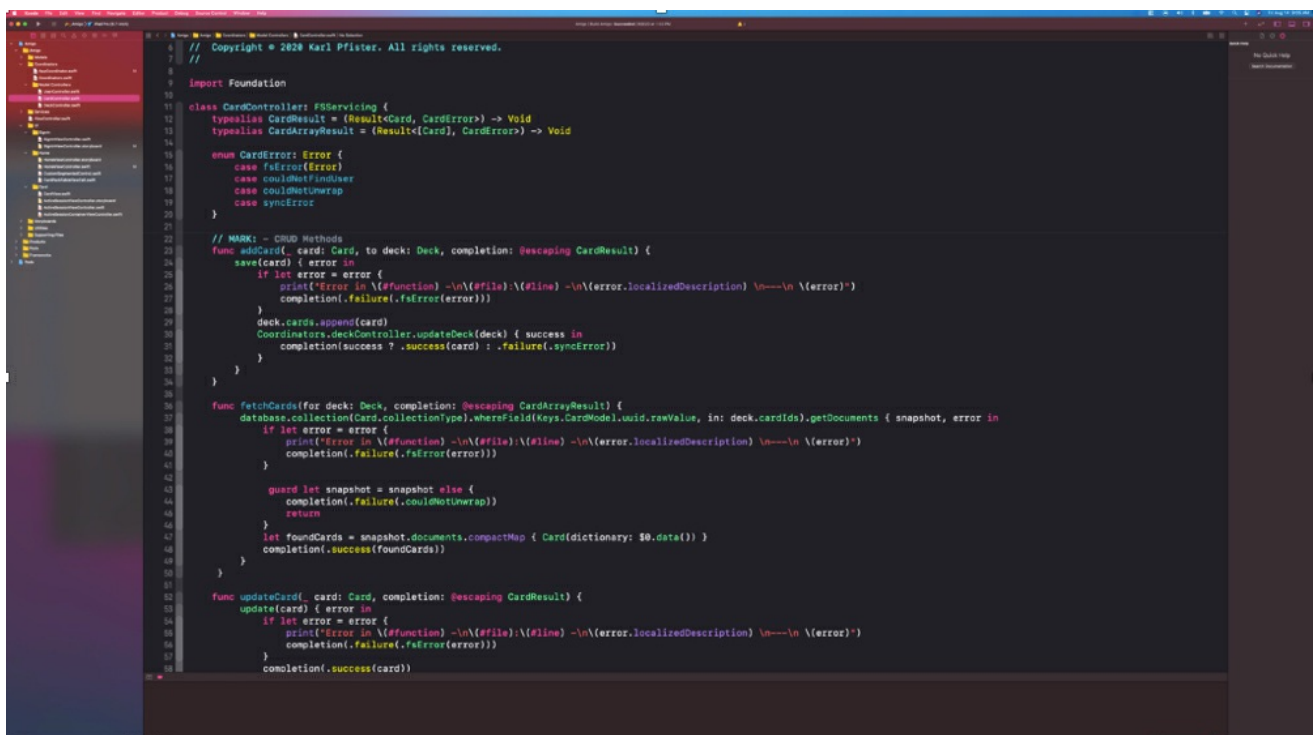
the top of the screen as Apple made room for a front-facing camera. Many engineers had to update their layout to make it display correctly. These are the kind of changes that iOS engineers anticipate every year. Some infographics show how the iPhone has evolved; it's fun to see how much has changed. When you build an iOS application or an app, as we like to call them, you get to do the work that may be handled by many individuals in a traditional software application.

iOS engineering includes:

- Building layouts and views like a web developer
- Creating interactions like a UX Designer
- Coding automation like a QA Engineer
- Creating and managing databases like a Software Engineer or Database Administrator
- Building network requests to send and receive data like a Back-end Developer or Network Engineer

2. Xcode and Swift

I primarily code in an IDE (Integrated Development Environment) named Xcode, using the programming language Swift, both of which were created by Apple, to work seamlessly together. The Xcode IDE includes a code editor, Interface Builder, a debugger, documentation, version control, tools to publish your app in the App Store, and even a device emulator. XCode contains everything you need to build iOS and Mac apps.



Swift Code in Xcode Application

When an app is developed leveraging the tools, programming language, and patterns for one platform, it is known as a native app. The app I've created for Poodle Jumper in Xcode using the Swift is specifically designed for iPhones. It can't be released on the Google Play Store for Android devices to download.

There are advantages and disadvantages to every mobile development method. It can be more expensive to

develop native apps, especially if you need to provide an iOS and Android version. At Poodle Jumper, we decided the performance benefits, accessibility, and integrations into Apple's ecosystem made a native app the right choice for us.

Swift provides a full programming language that can build programs for the Mac, Applications for iPhone, and iPad, Apple Watch, Apple TV, and you can even write server code with Swift! It's an incredibly powerful language, and you can make it do anything. Apple provides you with standard components to use, but if you want to make custom components, it is significantly more work.

Knowing the constraints for how Apple wants you to make apps is an important part of creating efficient code. This can make some UX Designers crazy when they want to step outside the box. Luckily, Mori (UX) is a big advocate for accessibility, and he knows that following the [Human Interface Guidelines](#) provided by Apple gives the app amazing accessibility with very little effort.



DID YOU KNOW

The “OS” part of iOS stands for operating system. An operating system is software that supports the basic functions of computers that we might take for granted. A login screen, the home page, navigation, even the clock. But have you ever wondered what the “i” in iOS, iPod, iPhone stands for? Apple first introduced the naming pattern with the iMac in 1998. At the time he explained the letter “i” was for internet, individual, instruct, inform, and inspire.



SUMMARY

This lesson explored the role of an iOS engineer, focusing on the **fast pace of change** in mobile technology and the tools and programming language used for iOS development. The primary language used, **Swift**, changes every year, which means that iOS engineers need to be continuously learning to keep up. They also need to adjust their views and code to match the new iOS feel with every new screen size, resolution, or change to device features. The lesson also discusses **Xcode**, an IDE created by Apple, which includes everything needed to build iOS and Mac apps, and the advantages of developing native apps specifically designed for iPhones.

Source: This tutorial was authored by DEVMOUNTAIN and Sophia Learning. Please see our [Terms of Use](#).