Lesson 1: Introduction to iOS Development and Xcode

INTRODUCTION

Welcome. This course covers the creation of iPhone and iPad applications. The course explores the development, distribution and submission process required in producing iOS applications. Attention to design and programming requirements for iOS development is also discussed.

To get started:

- Have a Mac computer with Lion or Mountain Lion operating system
- Install the iTunes application
- Register for a FREE developer account from Apple (requires e-mail address to setup)
- Download the Xcode application from Mac App Store

LESSON OBJECTIVES

By the end of this lesson, you will be able to:

1. Identify current mobile device iOS version.
2. Identify the features introduced in iOS versions.
3. Identify the different iOS devices including major features and hardware of each device.
4. Identify the operating system versions that work with different iPhone, iPod Touch, and iPad generations.
5. Identify limitations when developing iOS apps.
6. Identify advantages when developing iOS applications.
7. Identify the limitations of Xcode development tools (SDK and simulator).
8. Differentiate between native, hybrid and web-based applications including advantages and disadvantages of each.
9. Differentiate between the different levels of iOS Developer membership including cost, membership benefits and distribution characteristics.

LEARNING SEQUENCE

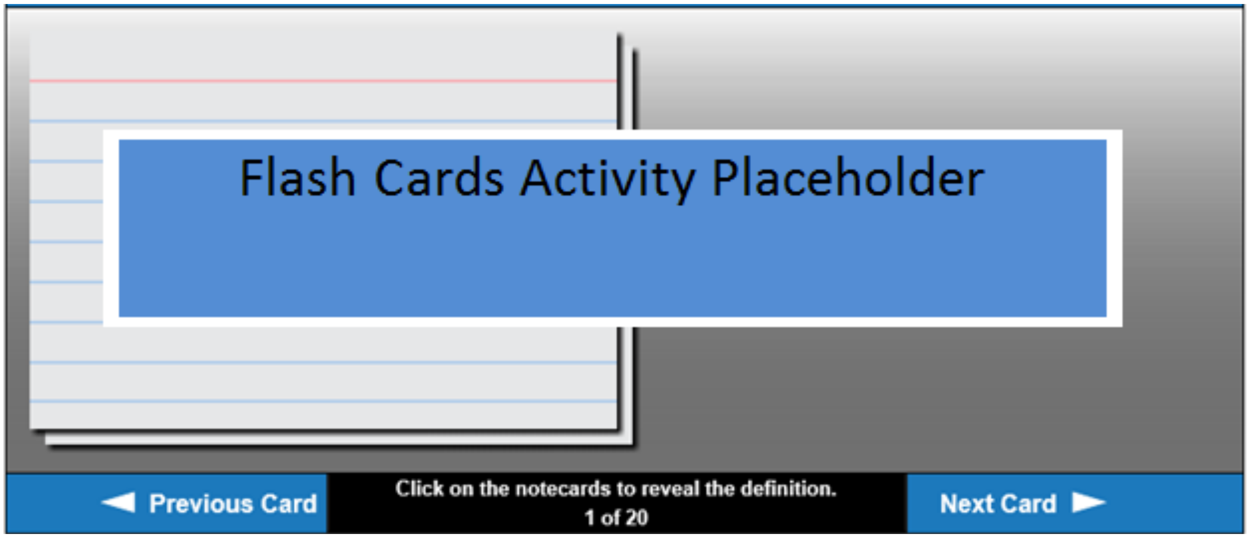
Required Reading	Read the following: • Lesson 1: Introduction to iOS Development and Xcode
Resources	Review the following: • Apple Developer Registration

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

	• iTunes Connect • iOS Devices • History or the iOS • iOS Versions • Siri • iOS 8 • iOS Developer Library
Assignments	Complete the following: 1. Register as Apple Developer 2. Practice Example – Howdy World 3. Quiz 1

KEY TERMS

As you read your lesson, pay close attention to the key terms and phrases listed throughout the lesson. These terms and concepts are important to your understanding of the information provided in the lesson.


Flash Cards Activity Placeholder

Click on the notecards to reveal the definition.

1 of 20

INSTRUCTION

Register as an Apple Developer

Registering as an Apple Developer provides free access to developer tools and resources including Xcode which is a requirement for the course.

Go to Register as an Apple Developer. Register with an existing Apple ID or choose to create a new, separate Apple ID for development. Review and agree to the Registered Apple Developer Agreement. Complete the profile. The choices made here will determine eligibility for the paid developer options. Registering as an Apple Developer is free, but there are two paid options available:

Standard Developer ($99/year)

- All features of a free developer
- Distribute apps through app store
- Able to test on actual devices

Enterprise Developer ($299/year)

- Develop for in-house use
- Allow team members to register under one lead developer
- Ability to distribute apps inside a company without going through the App Store

What is iOS?

iOS is the operating system for all Apple mobile devices and it is used in all versions of iPhone, iPad, and iPod touch.

Programming Languages

Objective-C is an **object-oriented programming language** and the primary programming language used by Apple for iOS. **Cocoa Touch** is the application programming interface (API) that provides a number of **frameworks**. Frameworks are technologies, services, and features that are used as building blocks when creating an app. Cocoa is the API used for the Mac OS. The difference is that Cocoa is more focused on using a mouse.

Objective-C started in the 1970's as a subset of C and was influenced by Smalltalk, an object-oriented, dynamically typed, reflective programming language that had an environment featuring development tools. NeXT Software licensed Objective-C which led to the development of NeXTSTEP, an object-oriented, multitasking operating system. Apple bought NeXT in 1996 and its OS X and iOS are descendants of NeXTSTEP.

iOS App Development

There is more to iOS app development than just the coding. There are also business considerations. For example, the demand for the app needs to be determined. Successful products are due to keen marketing strategies. There may also be financial decisions involved in the development.

Before the coding even begins, identify the audience for the app. Next, set goals and look at the overall design. Follow interface design principles to ensure that the app will make sense. Apps require usability testing so that users are not frustrated.

Quality control needs to be a point of focus throughout the process. Test at multiple steps along the way, not just at the end. Feedback from users will help keep the app current and provide information to help keep the app updated after it appears in the App Store. Be aware of any operating system updates because they can make the app obsolete. Apps will need to be updated at times when the operating system changes.

To successfully develop Apps, learn the lingo and focus on all things Apple. Subscribe to newsfeeds, appnews and other industry sources. It is helpful to have an iPhone, iPad or iPod Touch. Browse the iTunes App Store to investigate new apps. Pay particular attention to the reviews, both good and bad. Which apps are the most popular? Check out which Apps have won awards (and the reasons for it being an award-winning app). Note the apps that are featured in the App Store. Getting an app featured is a huge deal because it increases visibility, so look at what makes those apps special.

Get the Ball Rolling

After coming up with an idea for an app, go and see if it already exists in the App Store. If there are already similar apps available, what would make the proposed app different? If there are no similar apps, ask why that is? Is there just no demand for that app, or is it because the proposed app is truly unique? Research the idea to determine if it is viable.

Next, investigate what would be involved technically in creating the app. What are the hardware requirements? The proposed app may work on both the iPhone and iPad or it may be geared toward one or the other. Look at the hardware that must be on the device in order for the app to work.

A second consideration is the software requirements. What is the target iOS version for the app? Will there be any data and network requirements? For example, is the app pulling data from a database or perhaps writing to a database? What about going over a network?

Think ahead about how the app can be tested. There are companies that will test the app for a fee. Develop a strategy for handling bugs and keeping the app up-to-date. Maintaining the app and handling any problems that arise can require more time than the original development.

App Types

There are several different types of apps. This course focuses on native apps, or those that are built using Xcode with Cocoa Touch and Objective-C. A hybrid app is built in another language and then uses Cocoa Touch/Objective-C wrappers, a third-party program. The hybrid app is a mix of technologies that either encapsulates the code in a native wrapper or uses custom libraries. A Web-based app is hosted online and created with web technologies such as HTML 5, CSS, JQuery, or Sencha to name a few. This type of app does not go on the device itself; the app accesses a web location utilizing the Safari browser.

There are several advantages to web-based apps as well as disadvantages to be considered. The developer has total control of the development process since there is no App Store. Development and prototyping can be faster with the ability to correct bugs quickly. Also, this type of app can be developed for alternative platforms such as Android or Windows. Disadvantages include having to set up a payment system and limitations on the user interface effects and interaction with device hardware. The hybrid app and web-based app are beyond the scope of this course.

With native apps, it is easier to communicate with the hardware using Xcode for development. The iOS SDK offers a nice set of development tools to help. The App Store then handles the transaction including the purchase and distribution of the app. With the process put in place by Apple, the development cycle can be slower. Development requires a Mac and the ability to code in C or Objective-C. Finally, a native app is specific to iOS and cannot be ported to other OS based devices such as Android.

Overview of Building an App

Building an app involves three steps. As shown in Figure 1, the first step, Develop, includes setting up the development team, creating the project, developing the app, and publishing the app. Step 2, Test, is just that—testing, testing, and more testing. Step 3, Distribute, includes setting up iTunes Connect, submitting the app for approval, "shipping" the app, and responding to customer issues.

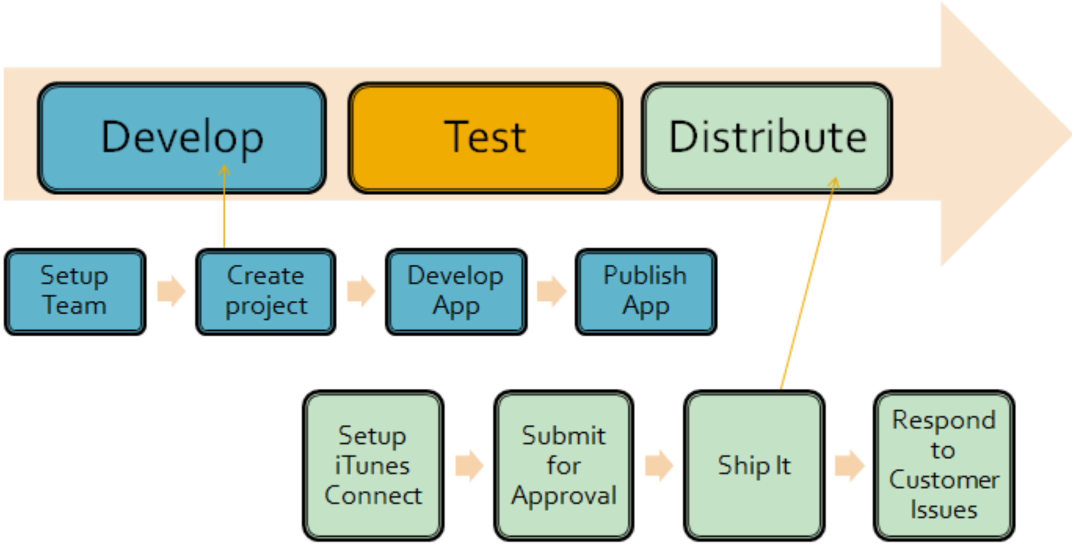


Figure 1: The steps followed to build an app

After registering as a developer, register for iTunes Connect. Read through and agree to Apple's distribution agreement. If a team will be working on app development, add team members and grant the appropriate administrative privileges. Certificates are available for both development and distribution. Register all iPhones, iPads, and/or iPod touches that will be used for development. Register application identifiers.

iOS Devices

iOS devices include the iPhone, iPad, iPad mini and iPod Touch. Review all of the iOS devices for a comprehensive look at devices, past and present, marketed by Apple Inc. This site provides an excellent comparison for all generations of each device including features and the highest supported operating system.

The iPod Touch is an inexpensive option for app development, if there is no iPhone available, since it is just like the iPhone without the phone feature. Since a camera was included on the iPod Touch starting with the 4G, the hardware is similar to the iPhone.

iOS Versions

Apple called its first operating system iPhone Software. With the release of the iPhone 3G came the iPhone OS. The iPhone 4 was the first phone to use iOS as its initial release operating system. A beta version of an upcoming update is available to developers before the official release. Beta versions are only available to developers with a paid membership.

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

It is important to note that hardware limits the iOS version that a device can run. For example, iOS 5.0 was the initial release operating system for the iPhone 4S. Look in the row that gives the highest supported operating system. This is important information because as the OS versions change, the features and the way that the code is written changes also. As a developer, backward compatibility may be one of the design goals.

The History of iOS

With the iPhone OS, the App Store became accessible and third party development became available. It is important to be familiar with the history of the iOS. iOS 3.x added copy and paste, messaging (MMS), a push notification server, accessory support, peer-to-peer support, a visual interface for creating data called Core Data and an iPad user interface (UI). The iPad UI included features such as popovers, split views, core text to name a few.

iOS 4 introduced a sort of multitasking that allowed the user to have more than one app running. This allowed one app to be in the foreground while the rest were in the background. iAd allows the addition of advertisements to an app. A high resolution screen was added along with retina display support. Games Center allowed games to share friends' lists and leaderboards. Other features included wireless printing, AirPlay, folders to hold apps, iBooks, playlists that could be created on the mobile device rather than copied over from the computer, and wireless keyboard support.

iOS 5 included iCloud storage and backup for syncing information, automatic reference counting (ARC) to handle memory management, storyboards to visually layout the iOS screens (or scenes) , newsstand support for subscriptions, Twitter integration and Siri. Additionally, the device did not have to be initially associated with a computer to start it up and load content. Visit Siri for more information on the personal assistant introduced by Apple Inc.

iOS 6 features added a social framework (i.e., Facebook integration). Passbook manages user coupons, tickets, etc. A Notification Center for Facebook and Tweets included a "Do not disturb" feature. Maps changed from Google Maps to Apple Maps, Apple's own map system. Siri features were added in addition to shared photo streams.

iOS 7 introduced an entirely new interface design with a different look and feel. A control center can now be opened with an up swipe to immediately access settings. Airdrop and iTunes music and radio were added. Multitasking gives the user the ability to visually swap between the current active app and the apps that are in the background. Finally, iOS 7 offered several new frameworks such as Text Kit, Sprite Kit, Game Controller, and Multipeer Connectivity Framework.

Apple introduced iOS 8 in the fall of 2014.

iOS Limitations

A developer can only work in one window with a fixed size and location. The screen size for the iPhone and iPod Touch is currently 320 x 480 pixels. With the iPhone 5, the screen size is 640 x 1136 pixels. The iPhone 6 has a screen size of 750 x 1334 pixels with the iPhone 6 Plus coming in at 1242 x 2208 pixels. There is limited access to the hardware and other parts of the device. There are no swap files on an iOS device, so available memory is limited to the memory on the device. Additionally, an app that drains the

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

battery is not going to be very popular. Reads and writes can only be done from the Sandbox. A Sandbox stores documents, preferences or other data. There is no administrative access similar to that available on a computer. Since a user may have to shut down the device quickly, a developer has to anticipate that there may be a limited response time involved. Remember, the developer uses touch and drag, not text and scroll, the norm for users of mobile devices.

iOS Advantages

There are several advantages of working specifically with iOS operating system. First, there is an extensive range of libraries and frameworks available. There is a set of robust, pre-built items that can be incorporated into an app. Secondly, the hardware and software standards are well defined. Since the developer is just dealing with Apple devices, it is easy to become familiar with the various features. The App Store has a step-by-step process which needs to be followed when submitting an app. Xcode is an integrated development toolset. Finally, a developer has the ability to take advantage of the hardware integrated within the device such as the **accelerometer**, **gyroscope**, cameras, GPS, etc.

The SDK

The software development kit (SDK) includes all of the interfaces, tools, and resources needed to develop iOS applications. There is a framework which is a shared library with the resources needed to support that library.

Xcode provides source editing, project management, documentation, a debugger, and an interface builder in one convenient tool. No secondary linking is needed to any tools. There is an iOS Simulator which provides feedback as to whether or not the code will build by simulating what the functionality would be. There are also instruments for performance analysis and a debugging tool. The iOS Developer Library is the repository for all of the aforementioned tools.

There are a few limitations of SDK. The simulator is based on Mac frameworks, not iPhone frameworks. As a result, the simulator may not be able to handle the iPhone app. The simulator is also missing some hardware features such as the accelerometer and camera because a computer would not have those features The Simulator Core location is fixed on Apple headquarters. This means that the developer may have to change the core location. The simulator is not able to deal with push notifications or keychain security nor does it process audio the same as an iOS device. Remember that the simulator is oriented more toward a MAC computer and not an iOS device.

Take this opportunity to browse the available resources on the iOS Developer Library website. Resources include guides, release note, sample code, technical notes and videos.

SUMMARY

This lesson took the student through the steps required to register as an Apple developer. Objective-C and Cocoa Touch were introduced. This lesson also provided an introduction to the various devices and iOS versions used by consumers. Finally, a developer needs to realize that there is a lot more that goes into developing an app than just the coding. There are various business decisions that may come into play in addition to issues involving quality control. The iOS Developer Library provides a wealth of resources and tools and is available for free.

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

ASSIGNMENTS

1. Register as an Apple Developer
2. Practice Example 1
3. Quiz 1

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands