General Assembly

MOB SYLLABUS

Course materials created by
Rudd Taylor, Tedi Konda, and Jeff Algera

BY THE END OF THIS COURSE

Students will be able to create an iOS app for iPhone or iPad that is App Store ready

UNIT 1: TRANSLATE WIREFRAMES INTO FUNCTIONAL APP INTERFACES

INTRODUCTION: XCODE, APPLE'S INTEGRATED DEVELOPMENT ENVIRONMENT

- Learn the basics of git and use GitHub to manage code base.
- Label all parts of the Xcode IDE
- Operate Xcode to create new projects and build interfaces using Storyboard

IOS APP CONTROL FLOW

- Describe the control flow of an iOS app
- Demonstrate how to extend an app to multiple screens
- Outline how elements are drawn on screen

PROGRAMMING BASICS WITH SWIFT

- Practice connecting interface builder to your Swift code
- Create a custom Swift function

UNIT 2: EXPERIMENT WITH OBJECT ORIENTED SWIFT TO ADD LOGIC TO IOS APPLICATIONS

LOGIC / COMPUTATIONAL THINKING WITH SWIFT

- Define computational thinking and translate instructions into basic pseudo code

OBJECT ORIENTED PROGRAMMING WITH SWIFT

- Object Oriented Programming with Swift
- Describe Object Oriented Programming
- Define MVC pattern as it relates to iOS app development: Show them sample code and have them label it M/V/C
- Utilize data structures to store multiple objects in an array and hash
- Programmatically create views.
- Apply Autoresizing and Autolayout for more flexible views and layouts.

APPLYING OOP TO IOS APPS

- Identify design patterns, such as delegation and notifications, to pass information throughout our apps
- Use Apple's Swift documentation to apply gestures to create interactive iOS apps

UNIT 3: BUILD APPS WITH PERSISTENT DATA AND REMOTE APIS

IOS FILES AND FILE I/O

- Discuss how iOS file system works
- Produce applications that store data across app sessions

IOS NETWORKING AND OPEN-SOURCE NETWORKING FRAMEWORKS

- Create iOS app network connections
- Describe how networking works at a lower level
- Describe AFNetworking's value, how it differs from iOS's built-in networking APIs

ADVANCED NETWORKING INTEGRATING OBJECTIVE-C AND SWIFT

- Explain how to use complex remote APIs and common Cocoa toolkits (using Swift/ObjC bridge)
- Integrate an arbitrary objective-C framework into a Swift project

UNIT 4: DESCRIBE THE IOS APP STORE SUBMISSION PROCESS

SUBMITTING YOUR APP TO THE APP STORE

- Utilize Xcode tools like Crash/Usage Tracking to optimize apps
- Navigate the app approval and distribution process
- Identify App store best practices and apply them to your final application