

iOS App Development with Swift

Instructor : Ali Salhi

Number of teaching hours : 54 divided by 18 Sessions (3 hours each).

Introduction:

This is a PBL course (Project Based Learning course), where concepts and learning topics will be introduced and explained by building different projects and apps. The course is designed to teach iOS programming through the creation of multiple apps using Swift, the iOS SDK, and Apple developer tools. Each project focuses on a single app or feature set.

The primary learning outcome for this course is that students will be able to design and create iOS apps. Students will leverage Swift, the iOS SDK, and Apple developer tools. With iOS as the platform, students will learn object-oriented programming, design patterns, type systems, functional language features, user interface design, best practices in programming, and problem analysis.

Upon successful completion of this course, students should be able to:

1. Define key programming terms relevant to Swift and iOS programming.
2. Describe the process of creating iOS apps.
3. State the purpose of the Apple developer tools, such as Xcode, Instruments, debugger, analyzer, and iOS Simulator.
4. Distinguish well-written code from poorly-written code.
5. Recognize patterns and idioms present in the Cocoa Touch API and other Apple frameworks.
6. Employ the Apple developer tools to create an iOS app.
7. Demonstrate programming best practices in Swift.
8. Examine and subdivide app functionality into properly designed components.
9. Explain and summarize iOS API features including location, mapping, sensors, gestures, multimedia and user interface components.
10. Plan, prepare and build an original iOS app, from concept to working program.

Prerequisites

Students are expected to have prior programming experience, such as programming with Java, C++, Objective-C, Python or Ruby, in an introductory programming knowledge is fine, Students are not required to have prior experience with Swift. This course assumes that students will engage in project-based, hands-on learning, either individually or in pairs, at an Apple workstation with the latest versions of OS X, Xcode and the iOS SDK installed prior to the first day of class.

Course Plan: (A different App/Project will be built in each Topic)

| Session | Section | Topic | Hrs | Details |
|---------|--|--|-----|---|
| 1+2+3+4 | Xcode Fundamentals and Swift. | Xcode, Interface Builder, AutoLayout and Swift | 3 | Familiarize yourself with Xcode, Interface Builder, AutoLayout and the iOS Simulator this include: <ul style="list-style-type: none"> • Running and modifying an iOS app. • Gaining a comfort level with Xcode. • Apply Auto Layout constraints to create adaptive user interfaces. • Discovering how to connect user interface controls to controller code. • Understanding the tools and technologies used to create iOS apps. |
| | | Fundamentals of Swift. | 9 | Learn the fundamentals of Swift which include: <ul style="list-style-type: none"> • Practicing the fundamentals of Swift syntax. • Practicing object-oriented programming with Swift. • Defining classes, working with objects and calling methods. • Discovering Swift data types and collections. • Analyzing code quality, and discovering advanced Swift topics. |
| 5+6+7 | Single View Applications and MVC. | MVC, Outlets, NSNotificationCenter, NSTimer and App Life Cycle Events. | 3 | Create a simple clock with outlets, NSNotificationCenter, NSTimer, and app lifecycle events. This include: <ul style="list-style-type: none"> • Applying Xcode, Interface Builder, views, models and controllers. • Assembling UILabel components • Defining controller outlets and the significance of IBOutlet. • Implementing Swift classes, properties and methods. • Identifying String objects, string formatting and the NSDate API. • Describing NSNotificationCenter and NSTimer. • Inferring application and controller life cycle events. |
| | | MVC, Buttons, Outlets, Actions | 3 | Create a stopwatch that displays the elapsed time, this include: |

| | | | | |
|-------------------------|-----------------------------|---|---|--|
| | | and NSTimer | | <ul style="list-style-type: none"> •Practicing applying Xcode, Interface Builder, and MVC. • Identifying UILabels, UIButtons, IBOutlets and IBActions. • Relating Model-View-Controller to project components. • Implementing Swift classes, properties and methods. • Applying String, string formatting, NSDate and NSTimeInterval. • Applying NSTimer, discovering Run Loops, assimilating lifecycle events. • Recognizing and assessing user accessibility. |
| | | MVC, UserDefaults, Picker Views, Delegates and Protocols. | 3 | <p>Build an app to display the the temperature in Fahrenheit while selecting a temperature in Celsius, which include:</p> <ul style="list-style-type: none"> • Practicing applying Xcode, Interface Builder, and MVC • Applying UILabel and UIPickerView components, IBOutlets and IBActions. • Implementing Swift classes, properties and methods. • Demonstrating Arrays, ranges and the map function. • Applying Model-View-Controller, and discovering view models • Describing protocols and delegates. • Using UserDefaults and property list files for persistence. |
| 8+9+ 10+11+ 12+13 | Frameworks and APIs. | WebKit | 4 | <p>Build a simple web browser using WebKit-and a UIWebView, which include:</p> <ul style="list-style-type: none"> • Defining URLs, requests, NSURL, NSURLRequest, and UIWebView. • Recognizing the UIApplication singleton. • Interpreting Interface Builder AutoLayout technology and constraints. • Combining constraints to ensure user interface usability on different screens. • Analyzing text-based input and different keyboard types. • Relating delegates, delegation and protocols to app implementation. |
| | | MapKit and Core Location | 4 | <p>Build a map app using MapKit and Core Location, which include:</p> <ul style="list-style-type: none"> • Describing how frameworks provide additional app functionality. • Defining URLs and the NSURL class. • Combining additional frameworks in an Xcode project configuration. • Relating delegates, delegation and protocols to app implementation. |

| | | | | |
|----------|---|--|---|--|
| | | | | <ul style="list-style-type: none"> • Relating Swift structures and classes. |
| | | AV Foundation Framework, Asynchronous Methods, “map”, and Closures | 4 | <p>Play four different sounds using the AVFoundation framework, which include:</p> <ul style="list-style-type: none"> • Integrating multimedia assets into an Xcode project. • Discover how an AVAudioPlayer object can play sounds. • Recognizing asynchronous method calls. • Assess repetitive code and efficient object instantiation. • Apply initializers and initialization requirements. • Practice using arrays, for-in loops, and the map function. |
| | | Gesture Recognizers (tap, pinch, rotate, swipe, shake) and Closures. | 3 | <p>Recognize taps, double taps, pinches, rotations, swipes and shakes, which include:</p> <ul style="list-style-type: none"> • Discovering how iOS encapsulates standard gesture recognition. • Combining view attributes and closures to exhibit visual effects. • Recognizing the Swift closure expression syntax. • Describing what closures are, and how they are invoked by other methods. • Exploring the features of the UIGestureRecognizer API. • Explaining the purpose of enumerations. |
| | | Touches and Core Graphics. | 3 | <p>Create a drawing using the touch-based user interface of iOS, and Core Graphics, Which include:</p> <ul style="list-style-type: none"> • Describing object-oriented inheritance and subclassing. • Discovering how to respond to touch events • Analyzing and developing a drawing strategy using touch locations. • Discovering Core Graphics contexts and procedural drawing idioms. • Distinguishing the intents of UIView and UIImageView objects. |
| 14+15+16 | Navigation, Tab Bar and Table View Controllers | Navigation Controllers and Segues | 3 | <p>Build an app that displays questions, answers, and a web search for more information, managed with a navigation controller, which include:</p> <ul style="list-style-type: none"> • Assembling multiple view controllers within an app storyboard. • Explaining how navigation controllers manage a controller hierarchy. • Practicing interface composition and establishing controller bindings. |

| | | | | |
|-------|------------------------|---|---|---|
| | | | | <ul style="list-style-type: none"> • Discovering how to use segues to manage controller relationships. • Recognizing the implicit behavior of navigation controllers. |
| | | Tab Bar Controllers, Asynchronous Network Requests, and JSON Data | 3 | <p>Use RSS data to display a top song, album, app and movie. Manage multiple controllers with a tab bar controller, which include:</p> <ul style="list-style-type: none"> • Assembling multiple view controllers within an app storyboard, and explaining how tab bar controllers manage view controllers. • Describing the starting procedure of an iOS app. • Analyzing app behavior and opportunities for code reuse. • Exploring RSS data and implementing network communication. • Applying asynchronous method invocations for long-running methods. |
| | | Table View Controllers, Navigation Controllers, and Segues | 3 | <p>Build a personal journal for recording important thoughts to demonstrate using table view controllers, which include:</p> <ul style="list-style-type: none"> •Assembling multiple view controllers within an app storyboard. • Explaining how table view controllers manage table view dependencies. • Recognizing the delegate pattern and relating delegation to protocols. • Discovering how navigation and table view controllers interoperate. • Assembling segues to encapsulate controller relationships. • Practicing analyzing model requirements to implement classes. |
| 17+18 | Selected Topics | Social APIs | 3 | Facebook+Twitter Apis integration. |
| | | Preparing App to publish | 3 | Publish an App to AppStore. |