iOS 11 App Development using *Swift*

Overview
This course is designed to train both novice and experienced developers how to create native iOS 11 Apps for iPhone, iPad and iPod using the ***Swift*** programming language. Created by experienced iOS developers, the course places a very strong emphasis on hands-on learning through a multitude of exercises and demonstrations. Students will be guided through a wide variety of in-class Labs specifically designed to offer practical solutions to common problems that address real-world production needs. Topics covered include: Multi-View Apps, Design-Patterns, Data-Persistence, Core Data, Web-Services, JSON and XML parsing, Maps and Location Services, working with Images and Animations, and handling Gestures.
An in-depth look at the **Xcode 9** IDE and its many features and tools will also be included.

Course Objectives
• Learn to use **Xcode 9**'s Interface Builder to design and build iOS App Interfaces.
• Utilize iOS Controls such as Buttons, Switches, Sliders, and Pickers.
• Build multi-screen Apps using Navigation Controllers, the Master-Detail ("Drill-Down") template, and the Tab-Bar Template.
• Data Persistence – save data on iOS devices using Text Files and Core Data for structured Databases
• Make calls to Web-Services from an iOS App
• Parse incoming JSON and XML data streams
• Work with Maps and Location Services to make location-aware Apps
• Enable Gesture Recognition to facilitate Swipes, Taps, Pinches, and Pans.
• Use Auto-Layout and Size Classes to properly configure Apps to support various screen sizes, device orientations (Portrait and Landscape) and iOS devices: iPhone SE's, 6's, and 7's, 8', iPads, etc.

Course Duration – 5 days

Prerequisites
• Basic knowledge of programming in ***Swift*** is recommended to get the most out of this class. (Note that a separate 3-day *Programming in **Swift*** class is also available.)
• Object Oriented Programming experience is recommended
• Basic familiarity with Mac computers and working in Mac OS X is recommended

Required Tools / Lab Setup

• Mac computers running **OS X 10.12** ("Sierra") or greater. An iOS device is optional.
• **Xcode 9** – free download from https://developer.apple.com/xcode/downloads/

Course Outline

I. **Your First App - "Prepare for Takeoff"**
   A. Creating a new Project in Xcode
   B. Designing an App interface using Interface Builder
   C. Working with the Objects Library and the Attributes Inspector
   D. Running and previewing the App using the iOS Simulator

II. **Adding Interactivity**
   A. Writing Swift code in the ViewController file
   B. Creating IBActions and IBOutlets
   C. Activating Controls: enabling Buttons, Switches and Sliders
   D. Using the Size Inspector
   E. Creating an Alert with action Buttons from code
   F. Working with Conditionals for basic Control-Flow

III. **Enabling User Input**
   A. Working with Textfields to read in user inputted data
   B. The iOS Keyboard and special alternate keyboard layouts
   C. Alternate methods for dismissing the Keyboard
   D. Using the Connections Inspector
   E. More Control-Flow

IV. **Working with Image Assets & App Icons**
   A. Creating a Custom Icon for your App
   B. Creating a Custom Splash Image for your App
   C. Managing Image Assets: Regular & Retina

V. **Writing Classes in Swift**
   A. Declaring a Swift class in a separate file
   B. Declaring Stored and Computed Properties
   C. Creating Instance Methods
   D. Control Flow with the Switch Statement

VI. **Using the PickerView**
   A. Single Component Pickers
   B. Multi-Component Pickers
   C. Working with the Date Picker and the **NSDate** class

VII. **Using Storyboards & Creating Multiview Applications**
   A. Using Navigation Controllers
   B. Working with Segues

C. Passing Data between View Controllers
D. Creating Bar-Tab Application

VIII. **Using TableViews**
A. Creating regular and Multi-Sectioned TableViews
B. Drill-Down menus
C. Master-Detail scheme
D. Creating custom TableView Cells

IX. **Using Collection Views**
A. Modifying Grid Layouts
B. Creating Custom CollectionView Cells
C. Enabling Cell-Tap Navigation to Detail Screens

X. **Creating Views from Code**
A. Initializing Objects with Code
B. Views, subviews, the Superview
C. The Subviews auto-array and View Index
D. Wiring up Views to IBActions using code

XI. **Detecting and Handling Gestures**
A. Swipe Gesture
B. Tap Gesture
C. Pinch Gesture
D. Pan Gesture
E. Rotation Gesture
F. Implementing Affine Transformations

XII. **Data Persistence**
A. Working with NSFileManager and the iOS File Directory
B. Saving Property Lists
C. Using Databases with the sqlite Library
D. Working with Core Data

XIII. **Maps, Core Location and Location Services**
A. Displaying User Location on Map
B. Creating Annotations & Pins
C. Switching Map Types
D. The MKMapViewDelegate Protocol

XIV. **Web Services**
A. Creating an NSURLSession
B. Making requests and queries to specific URL's
C. Reading returned XML & JSON data
D. Parsing incoming XML with NSXMLParser
E. Parsing incoming JSON Data

XV. **Integrating your App with Social Media**

A. Posting to Facebook and Twitter from within your App
B. Attaching Photos, URL's

XVI. **Universal Apps - Auto-Layouts, Handling Device Rotations, and Size Classes**

A. Working with Auto-Layouts
B. Implementing Size Classes
C. Handling Aspect Ratio Constraints

XVII. **Application Life Cycle & View States/Life Cycle**

A. Understanding Application States
B. Working with a ViewController's Life-Cycle
C. Creating Local Notifications

XVIII. **Unit Testing**

A. Using the XCTest Framework
B. Creating unique tests for specific funtionalities
C. Making Test Assertions
D. Creating performance-measurement tests

XIX. **Submitting to the App Store**

A. The iOS Developer's Center
B. Creating App Certificates, Identifiers and Profiles
C. Archiving the Project
D. Ad-Hoc and App-Store Distribution

***Note**: Clients are welcomed to request any other topics not appearing in this outline to be included in this course so that it meets their specific needs. Additional lessons can be created and substituted for existing ones so as to create an optimal training solution.*