Swift Programming for iPhone® and iPad® Applications

Duration: 5 Days

Price: $2595 *California residents and government employees call for pricing.

Discounts: We offer multiple discount options. Click here for more info.

Delivery Options: Attend face-to-face in the classroom, remote-live or on-demand streaming.

Students Will Learn

- Using Xcode to build iPhone and iPad apps
- Working with Swift data types
- Using Swift control structures such as if, while and for
- Creating and calling Swift functions
- Managing data using Swift arrays and dictionaries
- Designing and using Swift classes
- Understanding how ARC manages memory
- Handling run time errors in Swift
- Understanding iOS app life cycle
- Understanding Apple's implementation of the MVC design pattern
- Using Storyboards to design user interfaces
- Creating and configuring view controllers
- Building IBOутlets to interact with UI widgets
- Creating IBActions to handle UI events
- Using AutoLayout to create user interfaces for multiple iOS devices
- Implementing the Apple’s factory and delegate design patterns
- Creating master/detail apps
- Interacting with local files on the device
- Using Web services to manage data
- Persisting data locally using Core Data
- Capturing images using the camera
- Working with gesture recognizers
- Recording and playing back sound
- Interacting with Objective-C code
- Creating finite-length background tasks

Course Description

In this hands on Swift 3 programming course, attendees will learn how to develop iPhone and iPad apps using the Swift programming language and Xcode. Students will learn the Swift language and how to create Swift classes with properties, initializers and both instance and class methods.

Students will use Storyboards to design a user interface for iOS apps with multiple views. They will then configure view controller classes to interact with iOS views and controls (text fields, buttons, segmented controls, etc.) using IBOутlets, create event handlers using IBActions and then code events handlers.

Students will learn to persist data using three different strategies and implement each, read and write local files on the device, make asynchronous calls to Web services and parse XML data from the HTTP
response. Students will also use Core Data to interact with local SQLite databases.

The course also includes coverage of how to implement the Apple delegate design pattern which is used in many APIs including those interacting with the camera and recorder. Students will practice these skills by writing code to capture images with the camera and recording and then playing back, audio.

Students will learn how to utilize Objective-C code in Swift applications in order to take advantage of the rich variety of existing APIs and frameworks. The course also includes coverage of scheduled background execution to allow code to be run when the app isn't active.

Students learn how Apple’s Cocoa Touch UI Framework provides an abstraction layer of iOS for the iPhone and iPad. The course emphasizes safe programming practices. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency. Students will create Swift apps from the ground up, demonstrating the features of Swift and its supporting code libraries.

Course Prerequisites

Prior programming experience in an object-oriented language such as Objective-C, Java, C# or C++.

Course Overview

**Macintosh Programming Environment**

- Versions of iOS and Supported Devices
- Examining the macOS
- Exploring macOS Tools
- Enrolling in the Apple Developer Program

**Xcode IDE**

- Using Swift Playgrounds
- Creating Apps Using Xcode Templates
- Exploring the Xcode IDE
- Leveraging Xcode Debugging Support

**Swift Data Types**

- Declaring Variables and Constants
- Working with Swift Numeric Data Types
- Working with Strings
- Working with Dates
- Understanding Swift Optionals
- Leveraging Swift Generics
- Working with Collections (Arrays and Dictionaries)
- Using Tuples

**Swift Control Structures**

- Using Flow Control Statements (if, switch)
- Writing Loops (for, while, repeat)
- Writing and Calling Functions
  - Defining Parameters
  - Specifying Return Type
  - Using Named Parameters
  - Nesting Functions
- Using Swift Blocks

**Object Oriented Programming Constructs**

- Defining Classes
- Declaring Properties
- Writing Initialization Methods
- Creating Methods
- Understanding Public vs. Private
- Instantiating and Using Objects
- Understanding ARC (Automatic Reference Counting)
- Implementing the Delegate Design Pattern
  - Declaring Protocols
  - Implementing Protocols

**Swift Error Handling**

- Understanding Swift Error Handling
- Defining and Throwing Errors
- Propagating Errors Using Throwing Functions
- Handling Errors Using do-catch
- Understanding the Swift Error Type
- Converting Errors to Optionals
- Using defer to Specify Cleanup Actions
Optional Protocol Methods

**iOS Application Architecture and Design Patterns**
- Understanding iOS Application Architecture
- Exploring Design Patterns used in iOS Apps
  - Model-View-Controller
  - Singleton Pattern
  - Lazy Instantiation
  - Protocol/Delegate Pattern

**Introducing iOS Apps**
- Working with View Controllers
- Designing the UI Using Storyboards
- Building IBOutlets and IBActions
- Handling Events

**Enhancing iOS Apps**
- Exploring iOS Controls, Views and View Controllers
- Creating Multiple Views
- Using Segues for View Transitions
- Using Navigation Controllers
- Implementing Master/Detail Applications
- Using AutoLayout to Manage Different Size Devices

**Reading and Writing Files**
- iOS File System Structure
- Locating an Application's File Space
- Locating Files
- Creating Files
- Understanding Serializable Types
- Reading and Writing Files

**Working with Web Services**
- Using RESTful Web Services
- Creating Apps that Use Web Services
- Configuring and Using URLSession
- Working with Requests
- Formatting Data for Transmission Over a Network
- Parsing Data in Responses
- Using URLSessionTasks
- Configuring App Transport Security

**Using Core Data**
- The Managed Object Model
- Managed Object Context
- Creating a Core Data Application
- Understanding the Core Data Model
- Creating Entities and Attributes
- Subclassing NSManagedObject
- Fetching, Editing and Saving Core Data Objects
- Working with the Core Data Master/Detail Template

**Working with the Camera**
- Detecting the Camera
- Types of Media
- Working with UIImagePickerController
- Capturing and Processing the Image
- Saving the Image
- Configuring Camera and Photo Library Permissions

**Working with Audio**
- Using the AVAudioSession
- Recording Audio
- Playing Audio
- Configuring Microphone Permissions

**Mixing Swift and Objective-C**
- Understanding Mixed Language Applications
- Examining Objective-C Syntax
- Using Objective-C in Swift
- Using Swift in Objective-C

**System Events and Background Execution**
- Application Lifecycle Events
- Examining the App Delegate
- Reacting to System Events
- Running Tasks in the Background

_**iPad® and iPhone® are trademarks of Apple Inc., registered in the U.S. and other countries.**_