Swift Programming for iPhone® and iPad® Applications

Duration: 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

Price: $2495 *(Face-to-Face & Remote-Live)*, or $1495 *(On-Demand)*

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

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

Description

In this hands on Swift 4 programming course, attendees will learn how to develop iPhone and iPad apps using Swift and Xcode. Students begin by learning the fundamentals of the Swift language. They will explore how to build object-oriented applications by creating Swift classes with properties, initializers and both instance and class methods. They will see how to effectively use advanced Swift features like generics, closures, and error handling.

Students will use Storyboards to design user interfaces for iOS apps. They study how to configure view controller classes to interact with iOS views and controls (labels, text fields, buttons, segmented controls, switches, table views, etc.) using IBOutlets, create event handlers using IBActions and then code events handlers. They learn how to use segues to manage transitions between views.

Students will become proficient in implementing master/detail apps. Features explored include configuring table views, designing details views, implementing add features, and coding "swipe to delete". Students also learn how to implement different types of custom table view cells. This type of app is among the most common app found in the marketplace.

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

Students examine how to work with images, as well use touch and gesture recognizers to respond to complex user interactions like pinch to zoom. They learn how to use tab bar controllers to build a multi view app. They learn about the life cycle of an iOS app and how to write code to respond to state transitions, including scheduling code to run in background when the app isn't active.

Throughout the course, students work with the Apple's Cocoa Touch UI Framework. They

examine how to implement Apple's delegate design pattern which is used in many APIs. Students practice these skills by working with pickers and collection views, as well as writing code to capture images with the camera.

The course emphasizes best programming practices. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency. Students will create iOS apps using Swift from the ground up, demonstrating the features of Swift, iOS, and its supporting code libraries.

Prerequisites

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

Course Overview

Programming iOS Apps in Swift on a Mac

- What is iOS?
- Devices that Support iOS
- Building a Developer's Workstation
- 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
- Designing with Swift Generic Types
- 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 Function Types

Object Oriented Programming Constructs

- Defining Classes
- Declaring Properties
- Writing Initialization Methods
- Creating Methods
- Understanding Public vs. Private
- Instantiating and Using Objects

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

- Understanding ARC (Automatic Reference Counting)

iOS Design Patterns

- Working with Model-View-Controller
- Designing Singletons
- Using Lazy Initialization
- Implementing the Delegate Design Pattern
- Declaring Protocols
- Implementing Protocols
- Optional Protocol Methods

Enhancing iOS Apps

- Exploring iOS Controls, Views and View Controllers
- Creating Multiple Views Controllers
- Using Segues for View Transitions
- Using Navigation Controllers
- Using AutoLayout to Constrain Views and Manage Layout in Differently Size Devices

Designing Master/Detail Applications

- Implementing Master/Detail Applications
- Configuring Table Views
- Implementing Detail Views
- Implementing Add Item Functionality
- Implementing "Swipe to Delete"
- Supporting Multiple Types of Table View Cells

Working with Web Services

- Using RESTful Web Services
- Configuring and Using `URLSession`
- Working with HTTP Requests
- Formatting Data for POST Requests
- Designing Completion Handlers to Process HTTP Responses
- Parsing Data in HTTP Responses
- Configuring App Transport Security

- Converting Errors to Optionals
- Using `defer` to Specify Cleanup Actions

iOS Apps

- Understanding an Xcode Project Structure
- Designing the UI Using Storyboards
- Working with View Controllers
- Creating IBOutlets and IBActions
- Handling Events
- Specifying Different Types of Keyboards

Building Sophisticated User Interfaces

- Working with Labels, Buttons and Text Fields
- Using Switches and Sliders
- Allowing User Selection with Pickers
- Displaying Data Using Collection Views
- Using Tab Bar Controllers to Arrange Multiple Views

Reading and Writing Files

- iOS File System Structure
- Understanding an Application's Sandbox
- Locating Files
- Working With Serializable Types
- Reading and Writing Files

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

System Events and Background Execution

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

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

Touch Recognition and Gestures

- Understanding Multi-Touch Concepts
- iOS Recognizable Gestures
- Recognizing User's Touch
- Using Gesture Recognizers

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