Swift Programming for iPhone and iPad Applications

**Duration:** 5 Days

**Price:** £995

**Discounts:** For multiple course purchases, please [contact us](#) for applicable discounts.

**Delivery Options:** Perform training at your own pace via our [on-demand streaming](#) option or attend regularly scheduled live courses via [remote-live attendance](#).

### 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 the 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 IBOutlets to interact with UI widgets
- Creating IBActions to handle UI events
- Using AutoLayout to create user interfaces for multiple iOS devices
- Using Apple's singleton and delegate design patterns
- Creating Single View apps
- Creating Tab Bar apps
- Creating master/detail apps
- Customizing table views and table view cells
- Interacting with local files on the device
- Using Web services to manage data
- Persisting data locally using Core Data
- Working with collection views
- Capturing images using the camera
- Working with gesture recognizers
- Creating finite-length background tasks

### Course 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.

---

**Course 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
<table>
<thead>
<tr>
<th>Swift Data Types</th>
<th>Swift Control Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Declaring Variables and Constants</td>
<td>• Using Flow Control Statements (<code>if</code>, <code>switch</code>)</td>
</tr>
<tr>
<td>• Working with Swift Numeric Data Types</td>
<td>• Writing Loops (<code>for</code>, <code>while</code>, <code>repeat</code>)</td>
</tr>
<tr>
<td>• Working with Strings</td>
<td>• Writing and Calling Functions</td>
</tr>
<tr>
<td>• Working with Dates</td>
<td>• Defining Parameters</td>
</tr>
<tr>
<td>• Understanding Swift Optionals</td>
<td>• Specifying Return Type</td>
</tr>
<tr>
<td>• Designing with Swift Generic Types</td>
<td>• Using Named Parameters</td>
</tr>
<tr>
<td>• Working with Collections (Arrays and Dictionaries)</td>
<td>• Nesting Functions</td>
</tr>
<tr>
<td>• Using Tuples</td>
<td>• Using Swift Function Types</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object Oriented Programming Constructs</th>
<th>Swift Error Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Defining Classes</td>
<td>• Understanding Swift Error Handling</td>
</tr>
<tr>
<td>• Declaring Properties</td>
<td>• Defining and Throwing Errors</td>
</tr>
<tr>
<td>• Writing Initialization Methods</td>
<td>• Propagating Errors Using Throwing Functions</td>
</tr>
<tr>
<td>• Creating Methods</td>
<td>• Handling Errors Using <code>do-catch</code></td>
</tr>
<tr>
<td>• Understanding Public vs. Private</td>
<td>• Understanding the Swift Error Type</td>
</tr>
<tr>
<td>• Instantiating and Using Objects</td>
<td>• Converting Errors to Optionals</td>
</tr>
<tr>
<td>• Understanding ARC (Automatic Reference Counting)</td>
<td>• Using <code>defer</code> to Specify Cleanup Actions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iOS Design Patterns</th>
<th>iOS Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Working with Model-View-Controller</td>
<td>• Understanding an Xcode Project Structure</td>
</tr>
<tr>
<td>• Designing Singletons</td>
<td>• Designing the UI Using Storyboards</td>
</tr>
<tr>
<td>• Using Lazy Initialization</td>
<td>• Working with View Controllers</td>
</tr>
<tr>
<td>• Implementing the Delegate Design Pattern</td>
<td>• Creating IBOutlets and IBActions</td>
</tr>
<tr>
<td>• Declaring Protocols</td>
<td>• Handling Events</td>
</tr>
<tr>
<td>• Implementing Protocols</td>
<td>• Specifying Different Types of Keyboards</td>
</tr>
<tr>
<td>• Optional Protocol Methods</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enhancing iOS Apps</th>
<th>Building Sophisticated User Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Exploring iOS Controls, Views and View Controllers</td>
<td>• Working with Labels, Buttons and Text Fields</td>
</tr>
<tr>
<td>• Creating Multiple Views Controllers</td>
<td>• Using Switches and Sliders</td>
</tr>
<tr>
<td>• Using Segues for View Transitions</td>
<td>• Allowing User Selection with Pickers</td>
</tr>
<tr>
<td>• Using Navigation Controllers</td>
<td>• Displaying Data Using Collection Views</td>
</tr>
<tr>
<td>• Using AutoLayout to Constrain Views and Manage Layout in Differently Size Devices</td>
<td>• Using Tab Bar Controllers to Arrange Multiple Views</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Designing Master/Detail Applications</th>
<th>Reading and Writing Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Implementing Master/Detail Applications</td>
<td>• iOS File System Structure</td>
</tr>
<tr>
<td>• Configuring Table Views</td>
<td>• Understanding an Application's Sandbox</td>
</tr>
<tr>
<td>• Implementing Detail Views</td>
<td>• Locating Files</td>
</tr>
<tr>
<td></td>
<td>• Working With Serializable Types</td>
</tr>
</tbody>
</table>
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

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

Touch Recognition and Gestures

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

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

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