

Swift iOS 11: Idiomatic iOS 11 Development with Swift 4

Course Summary

Description

This two day class will introduce you to the concepts and workflow of creating modern Swift based iOS apps for the iPhone and iPad.

We begin day one with single-view applications. This lets us focus on what is different when writing code in Swift. You learn to communicate among elements of the GUI, source code, and model. We take advantage of advances in Xcode to position elements and support localization and accessibility. We end the first day with an exploration of multi-view applications and how to send information back and forth between views presented manually and those presented using storyboard segues. We separate the code into separate modules.

During the second day you will build more complicated applications with multiple scenes. We'll support table views and collection views that are presented using Navigation and Tab Bar Controllers. We look briefly at using Core Data, JSON, and asynchronous network requests.

Topics

- iOS App Fundamentals
- Multiple Scenes/h3>
- Table View Based Apps
- Network Applications (Optional)

Audience

This two day introductory course is for experienced developers who know Swift but are new to modern Swift-based iOS app development.

Prerequisites

There are no prerequisites for this course.

Duration

Two days

Swift iOS 11: Idiomatic iOS 11 Development with Swift 4

Course Outline

- I. iOS App Fundamentals**
 - A. Meet Xcode
 - 1. Create a new project
 - 2. Navigators
 - 3. Inspectors
 - 4. Console
 - 5. iOS Simulator
 - B. A simple UI
 - 1. The Storyboard
 - 2. Initial Scene
 - 3. Libraries
 - 4. Buttons and labels
 - 5. Attribute Inspector
 - 6. Constraints, Autolayout, and Stack Views
 - C. Outlets and Actions
 - 1. Create and configure an outlet and outlet collection
 - 2. Use the outlet and outlet collection
 - 3. Create and configure multiple actions
 - 4. Connection Inspector
 - 5. Implement the actions
 - D. Communication
 - 1. Model View Controller
 - 2. Understanding View Controllers
 - 3. Breaking up View Controllers
 - 4. MVVM
 - 5. Notifications
 - 6. Passing closures
 - 7. Simple Models
 - E. Custom UI (Optional)
 - 1. View Subclasses
 - 2. Targeting multiple devices
 - 3. Device specific layouts
 - 4. Localizing the Storyboard
 - 5. Localizing Strings
 - 6. Creating and using designables
 - 7. Inspectable properties
 - II. Multiple Scenes**
 - A. One App with Two Scenes
 - 1. Add view controller to storyboard
 - 2. Configure the second scene
 - 3. Create a button to trigger the segue
 - 4. Create and configure a segue
 - B. Manual Transitions**
 - 1. Custom view controller subclass
 - 2. Provide a public property
 - 3. Implement action to present second scene
 - 4. Create behavior to dismiss second scene
 - 5. Forward information
 - 6. Use the information
 - C. Segues**
 - 1. Connect Segue from first to second scene
 - 2. Implement prepareForSegue()
 - 3. Create Unwind Segues
 - 4. Create custom segue subclasses
 - 5. Pass information back and forth
 - D. Enhance the UI**
 - 1. Gesture Recognizers
 - 2. Image Views
 - 3. Asset Catalogue
 - 4. Animation
 - E. Modules**
 - 1. Separate Storyboards
 - 2. Storyboard References
 - 3. Add a framework
 - 4. Present content in a separate module
 - 5. Communicate information forward and back
 - 6. Minimize coupling
 - F. Delegates**
 - 1. Connect the delegate
 - 2. Conform to delegate protocol
 - 3. Closures as an alternative
- III. Table View Based Apps**
 - A. Separating Concerns
 - 1. New project using MVVM/MVC
 - 2. Gestures
 - 3. Complicating the model
 - 4. Using the Module for Insert
 - 5. Simple Persistence with User Defaults

Swiftly iOS 11: Idiomatic iOS 11 Development with Swift 4

Course Outline (cont'd)

- B. Table Views
 - 1. Delegate and Data Source
 - 2. Connecting to the same model with a different view model
 - 3. Enable Delete and Move
 - 4. Using the Module for Insert
- C. Assembling apps from pieces
 - 1. Notifications
 - 2. Tab Bar Controller
 - 3. Nav Controllers
 - 4. Connecting multiple scenes
- IV. Network Applications (Optional)**
 - A. Core Data
 - 1. Create the data model
 - 2. Generated classes
 - 3. The Core Data Stack
 - 4. The Context on the Main Queue
 - 5. Background Contexts
 - B. JSON
 - 1. Asynchronous Network Calls
 - 2. Handling Response
 - 3. Parsing JSON
 - 4. Creating Model object
 - 5. CRUD and Saving to Core Data
 - C. Custom Cells
 - 1. The Data Source and View Model
 - 2. Cell access to model instance
 - 3. Cell knowledge of view
 - 4. Filling the custom cell
 - D. Passing Info
 - 1. Prepare for segues
 - 2. Segue dependent set up
 - 3. Passing selected entity to next Controller
 - 4. Retrieving information for entity from Core Data
 - 5. Displaying the results
 - E. Collection Views
 - 1. Collection View Data Source
 - 2. Custom Cell
 - 3. Gathering relevant information
 - 4. Displaying the results