App Development with Swift

Data Collections

Swift

Learning Objectives

Earning the App Development with Swift Data Collections certification demonstrates the ability to transform creative ideas into functional applications. This comprehensive training program empowers learners to become skilled app developers using Swift®, Apple's programming language. Participants begin with essential Swift programming fundamentals and iOS development principles, then progress through industry-standard APIs that professional developers rely on daily.

Swift Language Usage

- **Protocols:** Describe purpose of protocols, adopt and define protocols, implement delegate pattern, conform to protocols with extensions, add default implementations.
- **Closures:** Define and use closures, pass closures as function arguments, use closures with sort/filter/map functions.
- **Extensions:** Add computed properties, methods, and initializers to existing types.
- **Swift Generics:** Understand benefits of generics, identify generic syntax, work with protocols and associated types.
- **JSON Decoding:** Read/write JSON data, use JSON serialization, serialize JSON into custom model objects.
- **Data Persistence:** Use Codable protocol for data serialization, access and manage Documents directory.

Debugging

- **Concurrency:** Understand iOS concurrency model, dispatch UI updates to main queue, handle networking responses.
- **HTTP and URL Sessions:** Print URL Session results to console, handle HTTP response codes and methods.
- **App Lifecycle:** Identify correct lifecycle events for app launch, background entry/exit.

Planning and Design

- **Model View Controller:** Understand MVC design pattern, role of view controllers, adapt MVC for external data sources.
- **UI Component Selection:** Choose appropriate use cases for table views vs collection views vs scroll views.
- **Animation Design:** Determine appropriate role of custom animations in apps.
- **Layout Planning:** Design complex layouts using compositional layouts, plan sections with distinct layouts.
- **User Experience:** Design complex input workflows, create forms with static table views.

App Development with Swift

Data Collections

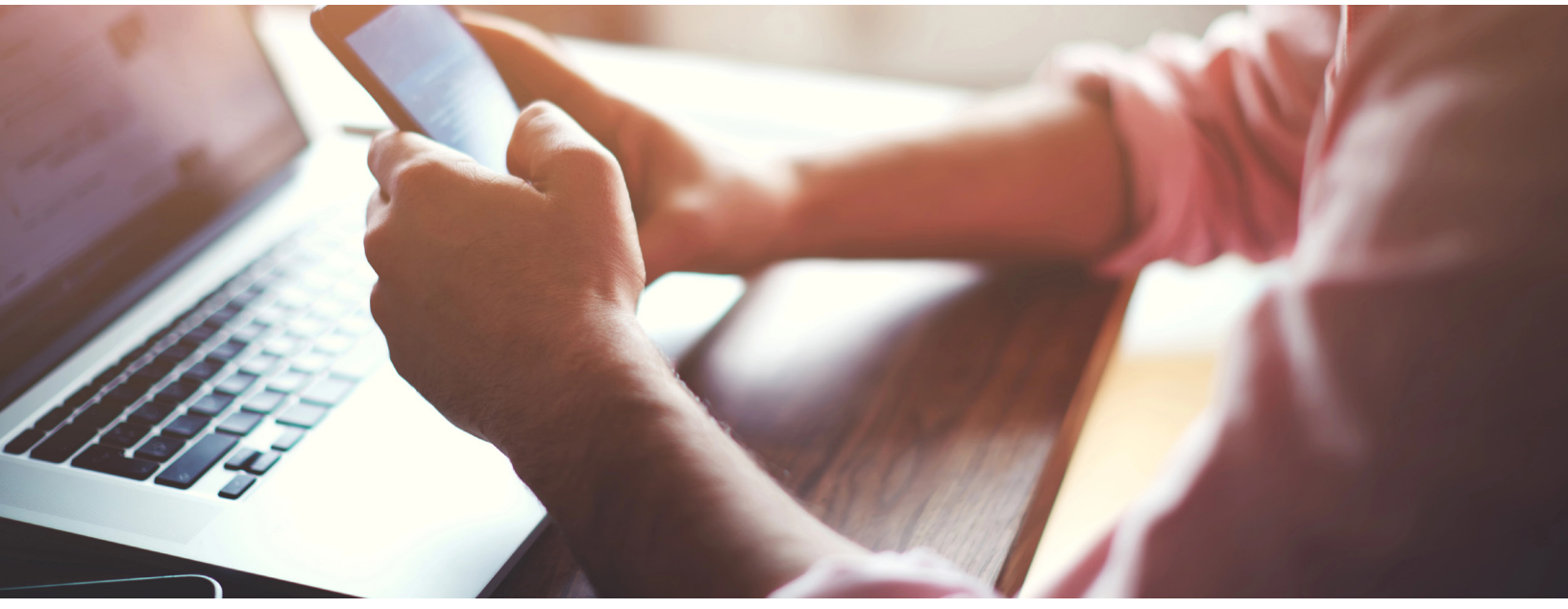
Xcode® Project Navigation

- **File Organization:** Organize project files using common MVC structure.
- **Interface Builder:** Create prototype cells, configure view controllers and system controllers.
- **View Controller Management:** Navigate between view controllers, pass data using protocols and delegates.
- **Project Configuration:** Configure apps for notifications, set up various system view controllers (alerts, activity, mail, image picker, Safari®).

View Building with SwiftUI®

- **View Positioning and Layout:** Use standard SwiftUI Views and modifiers to arrange or position a single View within a layout.
- **Multi-View Implementation & Navigation:** Build a multi-screen SwiftUI app that uses multiple Views to handle logic and user interaction.
- **List View Iteration:** Display collections of data using List Views.
- **View Structure Simplification:** Break down complex Views into smaller, reusable Subviews.
- **Data Sharing:** Enable communication between Views through @State, @Binding, @Environment, and Observable.

Swift



© 2025 NC3 | Safari, Swift, SwiftUI, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Swift and the Swift logo are trademarks of Apple Inc.