

**iOS Development using Objective-C and Xcode**

**Course Number:** MBL-136  
**Duration:** 5 days

**Overview**

This iOS Development using Objective-C and Xcode training class teaches developers how to develop and maintain native iOS applications in Objective-C for iPhone and iPad. Attendees learn basic and advanced skills in UI design and application development using Xcode.

**Prerequisites**

* Prior Object-Oriented Programming experience in Java, Objective-C, C#, or C++
* Mac OS usage experience

**Materials**

All iOS students receive comprehensive courseware.

**Software Needed on Each Student PC**

* Mac running the current or immediately previous version of macOS, with 8 GB RAM or more
* The latest version of Xcode (available for free from the Apple App Store)

**Objectives**

* Understand syntax, types, and flow control of Obj-C
* Perform Unit Testing with XCTest
* Integrate UI Design with Interface Builder
* Use blocks for passing code
* Utilize collections for storing data
* Understand protocols for defining roles
* Implement categories to extend classes
* Incorporate design Patterns to bind them all

**Outline**

* Introduction
* Getting Started
  + Intro to Obj-C
  + Data Types, Variables & Constants
  + Strings, Numbers, Bools
  + Literals
  + Type Casting
  + nil
* Unit Testing
  + Introduction
  + XCTest Framework
  + Asserts
  + Performance Testing
  + Xcode Service
* More Objective-C: Flow Control
  + Loops
  + Switch
  + Logical Operators
* UI with Interface Builder
  + Introduction
  + View Controllers
  + Views
  + Outlets
  + Actions
* More Objective-C: Functions
  + Parameters
  + Return Values
  + Blocks
  + Completion Handler
* Debugging
  + Introduction
  + Breakpoints
  + Debug Gauges
* More Objective-C: Objects and Memory
  + Properties
  + Dot Notation
  + Automatic Reference Counting (ARC)
  + Object Initialization
* Windows and Views
  + View Types
  + Responder Chain
  + View Resizing
  + Screen Size Considerations
  + Autosizing
  + Auto Layout
* More Objective-C: Collections
  + Arrays
  + Sets
  + Dictionaries
  + Enumerations
* Asset Management
  + Asset Catalogs
  + App Icon
* More Objective-C: Beyond Basics
  + Protocols
  + Categories
  + Blocks
* Application Patterns
  + Model-View-Controller (MVC)
  + Model-View-Presenter (MVP)
  + Model-View-View-Model (MVVM)
  + Target-Action Pattern
  + Subclassing
  + Delegation
  + Protocol Oriented Programming (POP)
* Storyboards
  + Introduction
  + Scenes
  + Segues
  + Moving Data Between Controllers
* Navigation Controller
  + Introduction
  + Root View Controller
  + Navigation
  + Segues and Passing Data
* UI Design
  + Constraints
  + Missing/Insufficient Constraints
  + Misplaced Views
  + Conflicting Constraints
  + Content Hugging, Compression Resistance
* Table Views
  + Overview
  + Delegates and Data Sources
  + Table Styles
  + Cell Styles
  + Prototype View Cells
  + Navigation
  + Static Table Views
* Universal Apps
  + Introduction
  + UI Considerations
  + Full Screen
  + UI Variants
* UIPicker View
* Directories and Files
  + Introduction
  + Obj-C Classes
  + Pathnames
  + Directories
  + Files
  + File I/O
  + UserDefaults
* CoreData
  + Introduction
  + Entities
  + Relationships
  + Code Generation
* Multitouch, Taps and Gestures
  + Touches
  + Gestures
  + Gesture Recognizers
* Drawing
  + Introduction
  + Points, Coordinates, Pixels
* Animation
  + CoreAnimation
  + Animation Blocks
  + Transformations
* App States
  + Introduction
  + AppDelegate
  + Considerations and Limitations
  + Background Execution
* Notifications
  + Overview
  + Permission
  + Local Notifications
  + Push Notifications
  + Notification Center
* CoreLocation
  + Basics
  + Location Accuracy
  + Distance Filter
  + Location Updates
  + Calculating Distances
* MapKit
  + Introduction
  + MKMapView
  + Regions
  + Map Type
  + Location
  + Annotations
* Concurrency
  + Introduction
  + Grand Central Dispatch (GCD)
* Networking
  + Reachability
  + Synchronous Downloads
  + Asynchronous Downloads
  + GET and POST Requests
  + JSON
* Localization
  + Introduction
  + Resources
  + Language and Region
  + Translation Considerations
* Running on a Physical Device
  + Developer Account
  + Development Certificate
  + Registering the Device
  + Build for Device
* Performance and Power Optimization
  + Introduction
  + Measuring Performance
  + Memory Considerations
  + Networking Considerations
* Deployment
  + Icons and Launch Storyboard
  + Archiving
  + Distribution
  + iTunes Connect
* Debugging
  + LLDB and Custom Breakpoints
  + Stack Trace
  + Thread Inspection
  + View Hierarchy Debugger
* API Design
  + Server calls
  + Completion Blocks
  + User Feedback
  + Handling UI Updates
  + Notification Center
* Instruments
  + Leaks
  + CPU/Performance
* Threading
  + Operation subclassing
  + Concurrent Queues
  + Serial Queues
  + OperationQueue vs GCD
  + Semaphore-based Locking
* Swift and Obj-C
  + Interoperability
  + Bridging Header
  + Swift from Obj-C
  + Obj-C from Swift
* Conclusion