

This course is aimed at developers wanting to learn how to create iPhone Apps. A solid foundation in Objective-C programming is recommended. Topics covered will include designing interfaces for iPhone Apps and exploration of the myriad GUI tools that come with Xcode - such as Buttons, Switches, Sliders, Page-Controllers, ImageViews, and many more. Students will gain an in-depth look at some of the built-in App Templates provided with Xcode - such as the Tab-Bar App, the Master-Detail (Drill-Down Menus) App, and the Page-Based (flip-book) App. In addition, students will get hands-on experience using the external libraries that provide additional functionality to Apps such as MapKit, SQLite3, MediaPlayer and CoreLocation.

Course Objectives:

- Write applications using the Objective C language and the Xcode IDE
- Build and use Tab Bars
- Use TableViews to display and manage lists of data
- Write apps that handle screen rotation and different device sizes
- Persist data using files and SQLite databases
- Perform graphical transformations and animation
- Write apps that interact with built-in iOS apps
- Interface with external services like Twitter and Facebook via web services
- Imbed maps and use location-based services in your apps
- Enable and implement In-App Purchase functionality
- Work with the iCloud
- Interact with device sensors such as the accelerometer

Audience: Developers and C-programmers.

Prerequisites: Good working knowledge of Object-Oriented programming using Objective-C.

Number of Days: 5 days

<p>1 Creating the first App – “Hello World” Exploring Xcode Working with Interface Builder The Objects Library View Hierarchy Creating a Custom Icon for your App Creating a Custom Splash/Launch screen for your App</p>	<p>3</p>	<p>Working with TextFields, Buttons, Labels, WebViews, PageControllers and more Understanding Views and Subviews Creating Views Purely from Code Mixing Methodologies Understanding View-Controllers & Application Templates The Single View template Exploring the App Delegate Creating and adding new View Controllers</p>
<p>2 Working with Outlets, Actions & Views Understanding the difference between Outlets and Actions</p>		

	Transitioning between Multiple View Controllers with Animations		Dynamically Adjusting Graphical Layouts in Response to Rotation
	Using the Tab-Bar Application Template	8	Persisting Data with Files
	Using the Master-Detail Application Template		Overview of the iOS Device File Structure
4	Working with the Keyboard		Reading and Writing Files to the Device
	Customizing the Keyboard for different Inputs		Creating and Modifying Property Lists
	Customizing TextField Behaviors	9	Working with Databases
	Methods for Dismissing the Keyboard		Importing the sqlite3 Library
	Using the NotificationCenter to Detect Keyboard Activities		Creating a Database, writing Tables, Inserting Records into Tables
	Using the ScrollView		Bundling a Database with your App
	Programmatically Responding to Keyboard Activities by Scrolling Views		Checking for Existence of Databases
5	Device Considerations – iPhone vs. iPad	10	Animations and Video
	Detecting Device Hardware Programmatically		How to use the NSTimer Class
	Dynamically Adjusting Graphical Layouts in Response to Device Type		Moving Objects across the screen
	Methods for Dealing with Multiple Devices		Using Transformations - Rotations, Scaling, and Translation
	Creating Universal App		Animating Arrays of Images
6	Using TableViews	11	Accessing Built-In iOS Apps
	Understanding the UITableView and UITableViewCell Classes		Using the Email Program
	The UITableView DataSource and Delegate		Launching Safari
	TableView/TableViewCell Properties: Header, Footer, Height, Fonts, Images, AccessoryTypes and more		Sending SMS messages
	The Master-Detail Template		Working with the Camera
	Drill-Down Menus and Navigation to other Views	12	Using Web-Services
	Data-Persistence using Property-Lists		Consuming a Web-Service in an App
	Creating Multi-Section Tables		Parsing XML
7	Supporting & Responding to Screen Rotations		Consuming and Parsing JSON Web Services
	Handling Device Rotation		Integrating Twitter
	Landscape Mode vs. Portrait Mode		Integrating Facebook
	Setting Preferred Device Orientation, Imposing no-Rotations	13	Working with Maps & Location Services
			Using the MapKit and UIMapView
			Getting & Displaying User Location
			Getting Directional Information
			Displaying Map Annotations
			Displaying Disclosure Buttons on Annotations
			Performing Reverse GeoCoding

- 14 **Working with iCloud**
 - Storing Documents in iCloud
 - Setting Project Entitlements
 - Managing iCloud Documents - the UIDocument Class
 - Storing Docs in iCloud
 - Storing Key-Value files in iCloud
- 15 **Working with the Accelerometer**
 - Using the Gyroscope and the Accelerometer
 - Outputting Sensor Data
 - Using the Shake API