

iOS DEVELOPMENT

Your First iOS App: From Zero to Your First Working App

Course Outline

Beginner Friendly



Course Overview

This course is the perfect starting point for anyone who has never built a mobile app before. You will learn how to set up your tools, understand the basics of Swift, Apple's beginner-friendly coding language and progressively build your very first iOS app step by step. No prior coding experience is needed. By the end of the course, you will have a fully working app running on your phone or the simulator.

Prerequisites

- A Mac computer running macOS Ventura or later
- Xcode (Apple's free app builder) installed from the App Store
- No coding experience required. We start from absolute zero
- Curiosity and a willingness to learn!

Skills You Will Learn

Set up Xcode and navigate its interface | Understand basic Swift concepts | Build simple app screens | Use buttons, labels, and images | Handle user input and show results | Run your app in the Simulator | Understand app structure and flow | Debug simple errors confidently | Store simple data in your app | Publish-ready app fundamentals

Weekly Breakdown

Week 1: Welcome to the World of iOS Development

Day 1: Setting Up Your Workspace

- SwiftUI vs UIKit: Why Apple switched and what it means for you
- Download and install Xcode from the Mac App Store
- A tour of the Xcode interface
- Creating your very first Xcode project
- Understanding what a 'simulator' is and how to use it

Day 2: Getting to Know Swift

- Variables(numbers, words, true/false), Data types: String, Int, Double, Bool
- Printing messages to the screen: Simple math and text in Swift
- Hands-on: Writing small Swift snippets in a Playground

Day 3: Making Decisions in Your App

- Loops and Conditionals: doing different things based on conditions
- Mini exercise: build a simple number-guessing logic

Weekly Project

Complete 10 Xcode Playground exercises covering variables, functions, and conditionals

Week 2: Building Your First Screen

Day 1: Laying Out Your App Screen

- Anatomy of ContentView.swift: what every line means
- Text, Image, Button: the three most-used SwiftUI views
- VStack, HStack, ZStack, Padding, spacing: arranging elements on screen
- Live Preview in Canvas: designing without running the app

Day 2: @State; Giving Your App Memory

- Why a UI without state is just a poster; Introducing @State and @State variables
- Toggle, Stepper, Slider: user input components wired to state
- Connecting button taps to logic using actions
- Hands-on: Build a live counter with increment/decrement buttons

Day 3: Lists & Dynamic Content

- List and ForEach: displaying arrays of data as scrollable rows
- The Identifiable protocol: why SwiftUI needs unique IDs
- Creating custom row views for a clean, readable codebase
- Section headers and grouped lists for polished layout

Weekly Project

Create an app with a big number in the middle and two buttons: one that increases the count and one that resets it to zero.

Week 3: Working with Multiple Screens

Day 1: Moving Between Screens

- NavigationStack, NavigationTitle, NavigationLink and Toolbar buttons
- Adding a second screen to your app

Day 2: Passing Information Between Screens

- Sending a name or number from screen 1 to screen 2
- Displaying the received data on the next screen
- Returning data back to the previous screen

Day 3: Tab Bars: Switching Sections of an App

- What is a tab bar? (Think: Instagram's bottom navigation)
- Adding a tab bar controller to your app, Setting icons and labels for each tab
- Organising your app into sections

Weekly Project

Build a two-screen app: a home screen with a 'View Profile' button and a profile screen that displays a name, photo, and fun fact the user typed in.

Week 4: User Input and Simple Data Storage

Day 1: Collecting User Input

- Text fields, Keyboards, Validating input, Switches, Sliders, and Pickers
- Hands-on: a simple signup form with name and age

Day 2: Alerts and Feedback

- Showing an alert message to the user, Creating action sheets, Activity indicators
- Haptic feedback
- Hands-on: Adding helpful error messages to your form

Day 3: Saving Data with UserDefaults

- What is UserDefaults? Saving a user's name or preference to UserDefaults
- Reading saved data when the app opens
- Hands-on: 'Remember Me' feature for a login screen

Weekly Project

Using UserDefaults, build a simple note-taking app where users can type a note, save it, and see it again the next time they open the app.

Week 5: Connecting to the Internet

Day 1: What is an API? Networking Fundamentals

- HTTP, JSON, and the request-response cycle, REST APIs explained
- The Codable protocol: automatically decoding JSON into Swift structs
- Handling nested JSON and optional fields gracefully

Day 2: URLSession & async/await

- URLSession, async/await
- Loading states, error states, and empty states in SwiftUI

Day 3: Building a Live Data App

- Designing your ViewModel for a network-powered app
- Showing a ProgressView while data loads
- Image loading from URLs with AsyncImage
- Hands-on: Live weather app using OpenWeatherMap free API

Weekly Project

Build an app that fetches data from a public API of your choice (weather, news, trivia, etc.)

Week 6: Polish, Test, and Ship

Day 1: Making Your App Look Great

- Consistent color schemes, Rounded corners, shadows, and card-style designs
- Adding animations, Supporting both light mode and dark mode

Day 2: Finding and Fixing Bugs

- Debugging and why every developer does it
- Using Xcode's built-in debugger to find problems

Day 3: Sharing Your App

- What is TestFlight? Understanding the App Store submission process
- Writing a good app name, description, and choosing keywords
- App privacy and what permissions mean
- What you have learned and where to go next

Capstone Project

Your capstone is your chance to combine everything you have learned into one original app of your choosing. Pick an idea that excites you, a to-do list, a birthday tracker, a recipe saver a quiz game and build it from scratch. You will plan your screens, write the code, style the interface, and present a working app. What you'll submit:

- A complete multi-screen app with at least 3 screens
- User input collected and saved with UserDefaults
- A polished, consistent visual design
- A short presentation (5 minutes) explaining your idea and how you built it
- Your Xcode project files submitted via the course portal

Meet Your Instructor



Prince Moses

Prince is a detail-oriented iOS Developer with a strong foundation in Swift and Apple's ecosystem backed by six plus years of professional UI/UX, Front-end development and Creative Design experience serving brands locally and internationally.