iOS – Sample Curriculum

Course Value Props

Build beautiful, fluid apps using Apple's new programming language Swift
Load your apps on your iPhone and learn the app store submission process
Understand the basics of managing the app development process

Pre-Work: TBD

Course Learning Goals

Unit 1: Translate wireframes into functional app interfaces

Introduction to XCode, Apple's integrated development environment

- Label all parts of the Xcode IDE
- Operate Xcode to create new projects and build interfaces using Storyboard

iOS app control flow

- Describe the control flow of an iOS app
- Demonstrate how to extend an app to multiple screens
- Outline how elements are drawn on screen

Programming Basics with Swift

- Practice connecting interface builder to your Swift code
- Create a custom swift function

Unit 2: Experiment with object oriented Swift to add logic to iOS applications

Logic / Computational Thinking With Swift

- Define computational thinking and translate instructions into basic pseudo code
- - Operate Git commands to manage code base.

Object Oriented Programming With Swift

- Describe Object Oriented Programming
- Define MVC pattern as it relates to iOS app development: Show them sample code and have them label it M/V/C
- Utilize data structures to store multiple objects in an array and hash
- Outline how elements are drawn on screen

Applying OOP to iOS Apps

- Use Apple's swift documentation to apply gestures to create interactive iOS apps

Unit 3: Build apps with persistent data and remote APIs.

iOS files and file I/O

- Discuss how iOS file system works
- Produce applications that store data across app sessions

iOS networking and open-source networking frameworks

- Create iOS app network connections
- Describe how networking works at a lower level
- Describe AFNetworking's value, how it differs from iOS's built-in networking APIs

Advanced networking Integrating Objective-C and Swift

- Explain how to use complex remote APIs and common Cocoa toolkits (using Swift/ObjC bridge)
- Integrate an arbitrary objective-C framework into a Swift project

Unit 4: Describe the iOS app store submission process

Submitting your app to the app store

- Utilize Xcode tools like crash/usage tracking to optimize apps
- Navigate the app approval and distribution process
- Identify App store best practices and apply them to your final application

Syllabus

Week 1: Getting Started, Intro to Nib / Storyboard

Pre-work: Design and UX principles for iOS apps.

Learning objectives: High-level introduction to programming languages/OO and how they are used, what is an app, define developer workflow and tools, get practice with interface builder

- Session 1
    - Dev environment setup, getting 'hello world' using storyboards on the phone
    - Use Xcode to create new projects and build interfaces through storyboard
- Session 2:
    - What is an app, how do apps run
    - Extend our app to multiple screens
    - Describe Object Oriented Programming, show students Swift and basic control-flow
    - Use Git and Github to create commits and pull requests on basic text files and Storyboards

Week 2: Intro to Swift, making Interface builder work with code

Learning objectives: Getting familiar with Swift, create an app with one screen and some functionality written in code

- Session 1:
    - Intro to Swift
    - Swift control flow
- Session 2:
    - How to tie interface builder to your Swift code
    - Swift sandboxes
    - The Cocoa touch documentation

Week 3: OO, UI elements and view controllers

Learning objectives: Understand the basics of objects, classes and protocols using view controllers and UI components as an example. Create a simple app with multiple screens.

- Session 1:
    - What's an object? What's a class? What are methods, variables and state?
    - Inheritance, polymorphism, encapsulation overviews
    - View Controllers and inheritance, view controller lifecycle
- Session 2:
    - What are class variables and methods?

    - UI controls, user interaction and gestures, inheritance continued

Week 4: Data structures, in-memory storage and more iOS design patterns

Learning objectives: Store data within app sessions, use basic iOS design patterns

- Session 1:
    - Data structures in iOS: Lists and dictionaries
- Session 2:
    - iOS watching patterns: KVO, delegate, notifications

Week 5: Views

Learning objectives: Understand how things are drawn on screen, be able to create and customize views in interface builder and code, understand 'springs and struts' layout as well as autolayout.

- Session 1:
    - Creating views in code
    - Springs and struts layout, view lifecycle and common methods
    - Hooking up IB to custom views
- Session 2:
    - Autolayout
    - Customizing views
    - Rotation

Week 6: In Class Lab - Putting it all together

Learning objectives: Practice building an app from single view app.

- *Both sessions: Review midterm project and work on it in class.*
- *Have students volunteer to present on a number of topics for the midterm, including: How they are laying out views, how they are keeping track of values in their calculator, how they are mapping buttons to actions*

Midterm Project:

Calculator

- Build a functional clone of the iOS calculator
- Calculator must be able to:
    - Add, multiply, subtract, divide
    - Handle integer and float input
    - Handle positive and negative input
    - Have a 'percent' button
    - Have 'clear' and 'all clear' buttons

Week 7: Files & Persistence

Learning objectives: Store data across app sessions, understand how files work

- Session 1:
    - How files work
    - How iOS handles files
    - Simple read/write exercises
- Session 2:
    - Writing to and reading from files
    - Temporary storage
    - User defaults
    - High-level overview of Core Data/ORM

Week 8: Networking

Learning objectives: Making simple network connections, understand how networking works at a lower level, be able to make connections from an iOS app

- Session 1:
    - Networking overview
    - How network connections work
    - Core iOS networking/reachability libraries
    - AFNetworking primer
- Session 2:
    - AFNetworking in depth

Week 9: Networking pt. 2, Cocoa/ObjC interaction

Learning objectives: Interacting with more complex remote APIs and common Cocoa toolkits (using Swift/ObjC bridge)

- Session 1:
    - How to use ObjC code in your Swift project
    - ObjC primer
- Session 2:
    - Using a complex ObjC client API (e.g. Facebook, 4sq, etc)

Week 10: Camera and Photos

Learning objectives: Familiarity with basic image processing, interacting with the camera and user photo library

- Session 1:
    - Images, image display and image storage
- Session 2:
    - Interaction with the camera and user photo library

Week 11: Review, app submission, getting an app production ready

Learning objectives: Be able to navigate the app approval process, review previous content

- Session 1:
    - App distribution
    - App store best practices
    - Crash/usage tracking
- Session 2:
    - Review of previous lessons

Week 12: Final project workshops

- Final project workshop, A/B testing services, logging
- Final project presentations, class wrap-up

Optional Content

Week XX: Maps and location

· *Learning objectives: Ability to add mapping and location-based functionality to an app*
· Maps views and displaying place locations
· User location, geolocation

Dropped Content

Week 7: Concurrency

· *Learning objectives: Perform resource-intensive work asynchronously to improve app user experience*
· Threads and the iOS concurrency model
· Grand Central Dispatch (GCD)

Week 8: Memory management and advanced debugging

· *Learning objectives: Debug complex memory issues, understand iOS memory management*
· ARC, the stack, the heap and the iOS memory management model
· Advanced debugging techniques, how to find memory leaks, sample performance optimizations

Week 9: Advanced drawing, UI elements and animations

- *Learning objectives: Draw custom UI elements and basic moving sprites*
- Custom UI element drawing, QuartzCore and Core Animation
- SpriteKit

Week 10: Putting it all together: algorithms overview and practice pt1

- *Learning objectives: Understand recursive and dynamic programming problem solving strategies*
- Greedy and memoized algorithms
- Recursion and traversal algorithms

Week 11: Putting it all together: algorithms overview and practice pt2

- *Learning objectives: Understand search and sort algorithms, when to use them*
- Search and sort algorithms