**Syllabus**

**Lesson 1: Getting Started**

* Ice breaker/materials
* Intro to iOS layout
* Outline developer workflow
* Creating your first project in Xcode
* Understanding the Xcode interface
* Using Git and GitHub
* *Bonus:* running your code on an iOS device
* *Assign homework 1*

**Lesson 2: Intro to Nibs/Storyboard**

* Define and implement nibs
* Storyboard multiple scenes
* Use navigation controller to link scenes
* Differentiate between Storyboards and nibs

**Lesson 3: Intro to Swift**

* Define Swift and its value to the iOS ecosystem
* Define and demonstrate playgrounds
* Define Swift’s fundamental data types
* Use variables and constants, and understand the difference between the two
* Apply optionals and understand when to use them
* Utilize control flow to create a simple program flow in playgrounds
* *Homework 1 due*
* *Assign homework 2*

**Lesson 4: Intro to Functions**

* Identify functions and implement best practices
* Be able to call and define functions that take parameters
* Be able to use the returned value from a function
* Understand what returning from a function does

**Lesson 5: Making Interface Builder Work with Code**

* Create hooks from interface builder to Swift code
* Create and implement custom classes
* Point out ability to access Xcode documentation for any external classes

**Lesson 6: Intro to Object-Oriented Programming**

* Define object-oriented programming
* Identify and Apply object-oriented principles: inheritance, polymorphism, encapsulation
* Differentiate between classes and structs
* Create protocols and apply them to classes, structs, and types
* *Homework 2 due*
* *Assign homework 3*

**Lesson 7: UI Elements and View Controllers**

* Demonstrate common UI elements and hook the IB elements to code
* Identify how to access documentation for IB elements and controllers
* Recognize and implement different view controllers
* Present views programmatically
* Identify arrays in Swift
* Introduce gestures
* Identify and implement methods associated with view controllers and UI elements inherited from their superclasses

**Lesson 8: Data Structures**

* Recognize the need for arrays and create/update arrays
* Identify array methods and properties
* Discover dictionaries
* Identify dictionary methods and properties
* Compare and contrast arrays and dictionaries
* *Bonus*: Apply closures through maps and filters
* *Homework 3 due*
* *Assign homework 4*

**Lesson 9: Hands-on with Table Views**

* Implement meeting a protocol in our apps, while getting a first look at the ‘delegate’ pattern
* Get hands-on practice with implementing table views and customizing table view cells
* Implement the passing of data from one screen to another

**Lesson 10: Using Basic iOS Design Patterns**

* Identify iOS design patterns and how they are used in our apps
* Define delegation and implement delegates in our apps
* Define notifications and show how to post and observe notifications
* Implement NSNotificationCenter notifications that already exist in our apps
* Identify best practices for using delegation vs. notifications

**Lesson 11: Introduction to Autolayout Using Interface Builder**

* Devise layouts relative to their superviews using Autolayout and interface builder
* Recognize view hierarchy and how views are constructed in our applications
* Define what a ‘constraint’ is, and how we apply those to views
* Distinguish very clearly the differences between frame and bounds
* Use CGGeometry framework to access the size of our views
* *Homework 4 due*
* *Assign homework 5*

**Lesson 12: Application Layouts with Autolayout**

* Design layouts with autolayout programmatically
* Use NSLayoutConstraints to set our constraints
* Debug autolayout errors and warnings
* Differentiate between autolayout and autoresizing
* Define, if not be able to use, VFL

**Lesson 13: Midterm, Day 1**

* The midterm tests students’ ability to translate wireframes into functional app interfaces as well as their ability to utilize Swift and the principles of object-oriented programming to add logic to iOS applications
* *Homework 5 due*
* *Assign midterm project*

**Lesson 14: Midterm, Day 2**

* The midterm tests students’ ability to translate wireframes into functional app interfaces as well as their ability to utilize Swift and the principles of object-oriented programming to add logic to iOS applications

**Lesson 15: Files & Persistence**

* Recognize the different types of persistence and their pros/cons
* Implement user defaults, but recognize that storage here should be limited
* Create property lists and save/read data from property lists
* Discover the iOS folder structure and where we should store certain types of files
* Use folder search within our applications to read and write files
* Create, read, and write to flat files
* *Assign homework 6*

**Lesson 16: Notsy & Core Data**

* Practice persisting data with plists, flat files, and user defaults
* Breakdown relational databases at a high level
* Breakdown the Core Data stack at a high level
* Create, read, update, delete (CRUD) data with Core Data
* Evaluate when to/not to use Core Data over plists/flat files/user defaults

**Lesson 17: Basic Networking**

* Recognize the benefits of JSON and REST
* Identify the benefits of using NSURLSession
* Evaluate and parse data returned in JSON format
* Integrate cocoa pods into our projects and use SwiftyJSON for better parsing
* *Homework 6 due*
* *Assign homework 7*

**Lesson 18: Modeling Data, Adding Back-Ends to Apps**

* Reorganize our data into a model
* *Bonus*: Integrate external services like Firebase or Parse to interact with back-ends for our applications
* *Introduce final project prompt and answer any questions related to the final project*

**Lesson 19: Review/Q&A/Lab Time\***

* If needed, go over any networking concepts you were unable to cover in lessons 17 and 18 and review any networking concepts that students might be struggling with
* *Homework 7 due*
* If you have time left over, use this as lab time where students can start discussing/working on their final projects and run their ideas by the instructional team

**Lesson 20: Lab Time\***

**Lesson 21: Instructor/Student Choice Framework Lesson**

* Teach a lesson on a subject not otherwise covered in the curriculum
* This subject can be either a topic that the instructor thinks would be helpful or a subject that students have expressed interest in
* You might consider conducting a poll of students on Slack, in order to gauge demand for various topics
* The lesson plan for lesson 21 contains a few ideas for frameworks you might consider teaching, as well as links to third-party resources that might be helpful
* If your class is still having trouble understanding any core concepts, you might consider instead using some or all of the lesson 21 class time to conduct a review lesson

**Lesson 22: Lab Time\***

**Lesson 23: App Store Submission Process**

* Identify the various components that a provisioning profile brings together
* Integrate a common beta-testing framework into a project
* Explain the concept of ‘beta’ and why it is important
* List the required assets for app submission
* List common reasons for app rejection

**Lesson 24: Final Project Presentations**

* Final Project Presentations
* “Lesson ∞” GA alumni presentation (delivered by producer)

\*For lessons denoted with an asterisk, lesson plans and slide decks are not provided. These are generally review lessons, lab lessons or project presentations.

\*\*Not all markets will schedule a lesson 0 (i.e. a class session that occurs before lesson 1). Please consult with your local producer regarding whether or not there will be a lesson 0 scheduled. If there is no lesson 0 in your market, some of the lesson 0 objectives can be rolled into lesson 1.