DRAFT as of 10 March 2015 – Not for official class use

CSCI S-65: Mobile Application Development Using Swift and iOS

Harvard University Summer School 2015

Instructor: Daniel Bromberg E-mail: daniel@basezen.com
Head TF: Austin Lin E-mail: austin@post.harvard.edu
Location: NW Science Bldg., Rm. B108 (52 Oxford St, 1/8 mi. north of Science Center)
Lectures: Mon, Wed 6:30-9:30PM (June 22 – Aug 15, 2015)
Sections: **TBD** (Intent: once weekly, 60 or 90 min, Thu or Fri)

OBJECTIVE: To lay sufficient foundation in all the tools and technologies necessary to create and deploy a realistic "App" (mobile software application) on latest-generation iPhones and iPads. We will focus on the fundamentals: the new **Swift** programming language; **Xcode** IDE navigation and the iOS development cycle; the essential **Foundation** and **Cocoa Touch** libraries; following Apple best practices, especially the Model-View-Controller paradigm; responsive event handling, especially gestures and network activity; the persisting and recall of user data; and finally, deploying apps to beta testers and ultimately to the public via the App Store.

While the programming libraries for these highly evolved devices are necessarily vast and impossible to cover in depth, we intend to show enough useful core APIs, examples, and over-arching patterns within, such that students will leave the class able to confidently apply their skills to the particular libraries (kits) required by their own future App developing efforts.

(The Apple iWatch is too new and will not be covered; however most skills should carry over.)

READINGS: Readings will come from a combination of: The Apple Developer Library; Chapters from new e-Books; and online tutorials. ***Students must understand that*** *due to the young age of the technology, having been released less than a year ago, written documentation is relatively scarce and academic-grade texts do not yet exist.* It is essential to be comfortable assembling know-how through a variety of sources, which will vary considerably in style, abstraction level, and intellectual depth.

The following introductory material is highly recommended to be done before the first class and will be due on the 2nd class:

https://developer.apple.com/swift/ **One-page promotional background**

https://developer.apple.com/videos/wwdc/2014/#402 **Official intro video at WWDC 2014**

https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/GuidedTour.html#//apple_ref/doc/uid/TP40014097-CH2-ID1
Technical introduction to Swift

The following is a sample of core readings from the iOS Developer Library, a source we will draw upon heavily. Note that while highly technical and a bit dry, the information is in clear and simple English and organized pedagogically.

View Controllers – the Glue of your iOS App (roughly 3rd week)
https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/AboutViewControllers/AboutViewControllers.html#//apple_ref/doc/uid/TP40007457-CH112-SW10

Model-View-Controller – the fundamental structure of your App code (roughly 3rd week)
https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html#//apple_ref/doc/uid/TP40010810-CH14

Physics tutorial – (roughly 6th week)
http://www.raywenderlich.com/76147/uikit-dynamics-tutorial-swift

EQUIPMENT: Students must own their own portable Mac running Yosemite (OSX 10.10) with at least 4GB of RAM, as well as an iPhone 5 or later running iOS8 (latest release recommended). Students should install the latest stable Xcode 6 before class starts and ensure it runs. For those interested in specializing their work for the iPad, they will need to own one as well. If students wish to deploy directly to the App store, they will need their own paid ($99/yr.) Apple iOS Developer account.

WORKLOAD: This is an intensive course with ambitious goals of teaching a new programming language, an IDE, several new frameworks/libraries, and a variety of UI and design concepts, and practical debugging and deployment knowledge. Assignments will be issued once per week and due the next week. Students must commit to roughly 15 hours per week of homework plus 7 hours per week of class.

GRADING: 25% Class participation (includes section); 50% general assignments; 25% final project

ATTENDANCE / ASSIGNMENT POLICY: On-time lecture and section attendance is required (see grading). Timely completion of reading and assignments is absolutely essential. There will not be time for the staff to help students catch up on missed work or missed classes. Late work will not be accepted except for medical emergency or religious observance (for the latter, prior notification required). General academic honesty policy may be found at **http://www.summer.harvard.edu/exams-grades-policies**.

ACCOMODATIONS: In accordance with **http://www.summer.harvard.edu/resources-activities/disability-services**

SCHEDULE ON NEXT PAGE

APPROXIMATE SCHEDULE: As below. Lectures will be **heavily** example-driven. Much of the material will not follow in precise segmentation as below but rather introduced and refined in layers over several classes, as theoretical asides to example code. For example, Xcode basics and project structure will begin on 6/24, covered heavily on 6/29, but will be revisited in pieces throughout the next 4-6 weeks as projects grow in complexity and students grow in comfort with the basics. **Schedule will also be adjusted to prevent average student workload from significantly exceeding aforementioned estimates. Second half of schedule may also adjust depending on students' interests, if practicable (e.g. if sufficient teaching materials available).**

Week	Date	Focus	Likely topics
1	6/22	Overview and Tech Intro	Policies; Student Surveys; Visual App demos; the Xcode/Swift Playground History of iOS & related; a look at Swift
1	6/24	Xcode and Swift basics	Creating & running a minimal Xcode project; in-depth Swift essentials (e.g. built-in types, functions, Optionals, Control Flow)
2	6/29	Structured Xcode and Swift	Organizational concepts: multiple-file projects & storyboards in Xcode; Swift **classes** & **struct**s; Running in simulator & essential debugging; References & Outlets; UIKit quick intro; Constraints & layouts intro
2	7/01	Cocoa Touch within Swift	Fleshing out Swift language data structures (classes, enums & structs); UIKit essential building blocks (interactive on-screen controls); MVC gentle intro: Views vs. View Controllers, delegation; layout as needed
3	7/06	Drawing and Gestures	Custom UIViews: 2D drawing; handling touch gestures; layouts as time allows
3	7/08	Model-View-Controller	Swift protocols; Observable properties; Delegation (as data source; as event handler); Separating the model; Navigation controller intro; Memory handling
4	7/13	Pulling pieces into multi-screen app	Navigation controllers, Segues; multiple views of same data, multi-screen apps; Flesh out Swift language features; Core Data intro
4	7/15	Persistence; structured data display	Objective-C foundation classes (NS*) and bridging to Swift; Property Lists; Core Data; Advanced UIKit: Table & List views
5	7/20	Web & Social Media	Storing & retrieving JSON/CRUD data; background data handling; Twitter & Facebook; the **Social** framework; advanced Tables;

			Images; background data cont'd
5	7/22	Development + Final Project Assigned	Flesh out areas as needed, esp. multi-screen apps; Debugging; Code refactoring; Swift subtleties. Introduce final project
6	7/27	Apps in real-time; Design	Interactive real-time peer-to-peer gaming & chatting; Apple Design standards; critique existing Apps
6	7/29	Project help	Take design challenges, API questions, & bugs, demonstrate most instructive ones to class
7	8/03	Cool Kits!	Last-minute project workshop: group feedback on "betas"; Physics (UIDynamics); moving objects (SpriteKit); GPS (CoreLocation);
7	8/05	Presentations & Party	2-minute presentation marathon; end-of-class celebration; possible informal awards based on class votes