Introduction to iOS Development with Swift
Spring 2016
January 19, 2016 - April 15, 2016
Tue/Wed/Thur (iOS)
Mon/Thu (Gainesville Dev Academy)
Location: CareerSource North Central Florida – Downtown Gainesville Career Center
Time: 6:00 PM – 9:30 PM

Professor: Jessica N. Jones, M.S.  
Email: jnj@ufl.edu

Teaching Assistant: L.T Carbonell  
Email: lt.carbonell@ufl.edu

Course Description: In this 12-week project-based course, students will learn to develop iOS applications using the Swift programming language and the Apple toolset. Additionally, students will learn basic concepts about designing intuitive and usable interfaces. On Mondays and Thursdays, students will hear lectures and presentations from the Gainesville Dev Academy. These lectures will cover topics including user experience, resume creation, interview preparation, agile software development and the use of various development tools. On Tuesdays and Wednesdays, students will learn the Swift language and iOS development concepts.

Course Learning Outcomes (From Apple’s Teaching App Development with Swift Course Overview): The primary learning outcome for this course is that students will be able to design and create iOS apps. Students will leverage Swift, the iOS SDK, and Apple developer tools. With iOS as the platform, students will learn object-oriented programming, design patterns, type systems, functional language features, user interface design, best practices in programming, and problem analysis.

Upon successful completion of this course, students should be able to:
1. Define key programming terms relevant to Swift and iOS programming.
2. Describe the process of creating iOS apps.
3. State the purpose of the Apple developer tools, such as Xcode, Instruments, debugger, analyzer, and iOS Simulator.
4. Distinguish well-written code from poorly written code.
5. Employ the Apple developer tools to create an iOS app.
7. Examine and subdivide app functionality into properly designed components.
8. Explain and summarize iOS API features including location, mapping, sensors, gestures, multimedia and user interface components.
9. Plan, prepare and build an original iOS app, from concept to working program.

Tentative List of Topics (From Apple project and lesson overviews):
• Running and modifying an iOS app
• Gaining a comfort level with Xcode
• Apply Auto Layout constraints to create adaptive user interfaces
• Discovering how to connect user interface controls to controller code
• Understanding the tools and technologies used to create iOS apps
• Practicing the fundamentals of Swift syntax
• Understanding object-oriented programming with Swift
• Discovering Swift data types and collections
• Applying UILabel and UIPickerView components, IBOutlets and IBActions
• Demonstrating Arrays, ranges and the map function
• Describing protocols and delegates
• Using NSUserDefaults and property list files for persistence
• Describing object-oriented inheritance and subclassing
• Discovering how to respond to touch events
• Practicing establishing connections between controllers and views
• Describing how frameworks provide additional app functionality
• Defining URLs and the NSURL class
• Combining additional frameworks in an Xcode project configuration
• Discovering the fundamental features of the MKMapView API

Course Materials:
All students enrolled in this course are encouraged to have:
• A Mac computer that they can bring to class everyday, though one can be provided to those in need (contact CareerSource North Central Florida or Gainesville Dev Academy)
• The latest non-beta version of Xcode installed on the Mac laptop that they bring to class. Xcode can be downloaded at the following link (https://developer.apple.com/xcode/download/)
• An Apple ID. Students can create an Apple ID by visiting the following website (https://appleid.apple.com/)

Grading:
• 55% -------- Programming Assignments
• 5% -------- Class Participation
• 40% -------- Final Group Project
  o 5% -------- Final Project Pitch
  o 5% -------- Final Project Updates (2)
  o 15% -------- Final Project Submission
  o 5% -------- Final Project Presentation
  o 10% -------- Group Evaluations

**No Final Exam

Programming Assignments: At the end of a section, students will be given a programming and/or writing assignment that will be due in at least one week.

Group Project Pitch: Students will be placed in groups. Each group will propose a project on which the group will work throughout the semester. The pitch can be a written document or video presentation that explains the proposed project, each group member’s role and their projected progress for each of the two (2) project updates. NOTE: The requirements for this assignment are subject to change.
Project Updates: Throughout the semester, students will submit two (2) project updates. For each update, each group will present a written document or video presentation that compares their actual progress to their projected progress. Groups will describe what they were able to accomplish, what they did not accomplish, problems they faced and, if necessary, how they will get back on track (individually and as a group).

Group Evaluations: For each project deliverable, every individual will rate themselves and each group member on their contribution to the team. Group members will be given a rating of 0 (unacceptable) to 4 (excellent) for each of the following areas:

- Quality of submitted work products
- Communication with the team
- Timeliness of contributions
- Willingness to work as a member of the team
- Completion of non-development related tasks

The evaluations for each group member will be compiled and averaged. The student’s project evaluation grade will be equivalent to this average.

Final Project Submission: Each group will submit their completed project including all code and assets. Additionally, each group will submit a technical manual and a user manual.

Final Project Presentation: Each group will give a presentation that describes their final project.

Grading Scale

<table>
<thead>
<tr>
<th>Score ( Rounded to the nearest point)</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 - 90</td>
<td>A</td>
</tr>
<tr>
<td>89 - 80</td>
<td>B</td>
</tr>
<tr>
<td>79 - 70</td>
<td>C</td>
</tr>
<tr>
<td>69 - 60</td>
<td>D</td>
</tr>
<tr>
<td>&lt; 60</td>
<td>F</td>
</tr>
</tbody>
</table>

NOTE: Students must receive a grade of 70% or better to receive a certificate of completion.

Policy on Late Submissions:
A 20% PENALTY will be assessed for any assignment submitted after the announced due date. An additional 10% penalty will be assessed for each day thereafter.

Projects That Do Not Build:
Any project that does not compile will receive a 40% PENALTY. Students will be notified that their project does not build and will have 24 hours from the time a “Delivered Receipt” is received to resubmit the assignment. If the assignment is not resubmitted within the 24-hour window, the student will receive a 0.

Policy on Extension Requests:
All extensions must be requested in writing at least three (3) days before the announced due date. In your request, please provide the assignment name, the original due date of the assignment, your suggested due date and the reason for your request.

**Tentative Schedule**
1. Syllabus and Class Expectations / Intro to Project Management
2. The Swift Programming Language / Intro to User Experience
3. iOS App Basics / Git 1, Intro to GitHub
4. Segues / Git 2, Intro to Tools
5. Tables and Persistent Storage / Apiary, Intro to APIs, Postman
6. Online Storage / Intro to TDD, XCode Tips and Tricks
7. Social Network Integration / Charles, Fabric
8. Sounds, Gestures and the Camera / UI Testing, Unit Testing
9. Maps and Geolocation / Postman / Accessibility, iOS 9 HIG
10. The Web / Intro to TDD / Git 3
11. Device Sensors / Distribution
12. Games with SpriteKit
13. Calendars, Address Book, and Email

**Important Dates:**
- January 18: Martin Luther King, Jr. Day (No Class)
- February 29 – March 4: Spring Break (No Class)

**Final Project Due Dates:**
- Groups Assigned: February 4
- Project Proposal: February 25
- Project Update 1: March 17
- Project Update 2: March 31
- Final Presentations: April 15

**Accommodation for Students with Disabilities:** Students requesting classroom accommodation should inform the instructor as soon as possible.

**Honor Code and Collaboration:** Students MUST document all reused code. Failure to document code found online or provided by other students WILL be viewed as plagiarism AND WILL RESULT IN A 0 FOR THE ASSIGNMENT.

NOTE: This syllabus may change depending on the pacing of the class. Any changes will be beneficial to students and will have minimal impact to student grades, the grading categories and to the student workload.