Group 6(Dhruval, Shaluvi, Francis, Noor, Marcus)

Date: 3.20.16

Course: SWE 4743

Instructor: Dr. Jones

A *Swift* Bibliography

1. <https://swift.org/> (D-Val)  
    The Swift website was very helpful in learning about Swift as a new open-source programming language and also finding the best resources to learn Swift. The website details the goals of what Swift attempts to accomplish in clear details, such as its emphasis on performance, features, and continuity. The website also provides a one stop shop for downloading releases, documentation, and even source code. Another aspect of the website is it’s support for bringing together the Swift community, which allows developers to gather and bounce ideas off of each other. Since Swift is an open source programming language, the website tracks, supports and documents how developers can add to Swift via GitHub continuous integration. This truly is the home of Swift.
2. <https://developer.apple.com/swift/> (D-Val)  
    Apple provides a neat blog for Swift with various resources. They describe the latest and greatest updates for the language, as well as how to implement features. They also provide resources and guides so that developers can learn Swift interactively. This is very helpful for new developers. The only requirement is to use Xcode and Xcode Playground. The resources extend from IDE’s, How-To’s, videos, blogs, forums, etc. An absolute wonderland of information.
3. <http://tnw.to/d4l1C>   
    This link is to an interview held between thenextweb.com and Apple’s Senior Vice President of Software Engineering Craig Federighi. The interview focuses on why swift is becoming open source and what that means for the language and company. It also goes over the future of development for both Swift and Objective-C. This interview is key because it helps us hear straight from the top of Apple’s software engineering branch what the future of Swift is and where Apple wants the language to go. It also shines a spotlight on what the future is of Swift and Windows support as the language opens it’s doors and becomes an open source language. Judging from this article the next interview with Craig Federighi when Swift 3.0 comes out will hold even bigger goals for the language..
4. <https://en.wikipedia.org/wiki/Swift_(programming_language)>   
    This wikipedia article on the language is a great starting point for research into the language. We find basic information as when it first appeared, who the developer is and the languages that influenced swift. After the first introduction we see the coverage of the language’s architecture and its similarities to C, Objective-C and its differences from Objective-C. This page was very helpful in learning what pathways would need more research and where there would be confusion. The language is very young which is why there aren’t many large sources that can easily and quickly convey information on the language. The vast coverage of information by Wikipedia also helps as an individual can quickly jump to other pages as well.
5. <http://code.tutsplus.com/tutorials/protocol-oriented-programming-in-swift-2--cms-24979>  
   The code.tutsplus website is about the swift 2. It talks about how Apple added a range of new features and capabilities to the swift programming language. This website explains the basics of protocol-oriented programming in swift and how it differs from object-oriented programming. It also talks about the swift standard library and it allows us to add properties or functions to existing protocols. For default behaviors to protocol extensions, protocol-oriented programming in swift is going to be adopted by many future API’s and will completely change the way in which we think about software development.
6. [https://www.weheartswift.com/object-oriented-programming-sw](https://www.weheartswift.com/object-oriented-programming-swift/)  
    The weheartswift website explains the Apple’s frameworks in object-oriented programming. It gave us the overview of the swift language and also explains about the 3 object-oriented programming concepts such as encapsulation, polymorphism, and inheritance. This website explains the fundamental concepts of the Object-oriented language. It gives us an explanation of the classes and objects in swift with the code example. It talks about classes and instances, where they can have associated values named properties with code example. It also talks about the methods, class properties, inheritance, overriding, protocols, polymorphism, and view objects.
7. <https://cocoapods.org/pods/SwiftMongoDB> (MS)  
    This website was used to gain an understanding of how the MongoDB can be used by Swift. It provided examples of how to use the wrapper class to allow swift developers to create schemas and query the database.

1. <http://www.theappguruz.com/blog/use-sqlite-database-swift>  
    Gives examples of how to use the FlyingMeatDatabase as a wrapper for the SQLite class. The most important points were about how to link the SQLite with a project, how to solve the ARC (Automatic Reference Counting) problem and bridging Objective-C files to Swift.
2. <https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html#>  
    Apple gave the most details about the inner workings of the language from simple variable declarations to more complex bitshifting. It was the most comprehensive source for learning swift. The site went over syntax, code demonstrations and some best practices.While certain sections, such as closures and protocols were not as clear and easy to understand as other sections like functions and deinitialization, overall it was a valued source of information about the language in general.
3. <http://codewithchris.com/iphone-app-connect-to-mysql-database/#connectiphonetomysql>  
    This site explained the concept that devices typically do not directly communicate to a database for security reasons and the standard way is to use an API to be a middle man between the data from the outside world. The author made MySQL database hosted on a local server via phpmyadmin, created an dAPI to facilitate the communication between client and server.
4. <https://www.perfect.org/about.html>  
    Perfect is a frameworks that allows the creation of web and REST service in swift. The source further details benefits such as reduced code, full Xcode support and the entire code base being written in a single language. Supports JSON encoding and decoding, cURL and also LibEvent.
5. <https://developer.apple.com/library/mac/documentation/Foundation/Reference/NSURLSession_class/index.html#//apple_ref/occ/cl/NSURLSession>  
    This site provided good information about how iOS in general handle URL connections. It explained the URLSession, URLConnection returning data through delegates and callbacks. It also provided explanations for making POST requests, authentication and working with web services through parsing and JSON Serialization.