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

Assignment Prompt - Week 2

1) Project prompt
Complete all TODOs in the provided app. The provided app contains a series of exercises going over Swift basics and control flow. By completing these exercises you will gain confidence and familiarity with variables, functions, classes and control flow in Swift. You will also have hands-on experience with XCode and other common iOS development tools.

2) Goals of project

- Be able to read and write basic Swift statements and control flow.
- Know how to navigate XCode in order to quickly prototype your code.
- Be able to use and understand common programming terminology (e.g. ‘variable’, ‘method’, ‘class’).
- Understand how to make hooks from interface builder into your Swift code.
- Understand the basics of view controllers and how they impact your apps.
- Be able to pull down and submit code via Github

3) Project requirements
Your app must:

- Successfully meet all tasks outlined in the given app.
    - Format: Sample app has all view and view controller TODOs filled out and functioning.
- TODO one asks the user to hook up a button in interface builder to a new function (to be written) in a view controller subclass. When that button is clicked, the function to be written must make a label say ‘hello world!’
- TODO two asks the user to modify the first input by making a connection from ‘name’ and ‘age’ text boxes defined in interface builder to their class. They must look at the string entered in the text box and print out “Hello {name}, you are {age} years old!”
- TODO three asks the user to print “You can drink” below the above text if the user is above 21. If they are above 18, print “you can vote”. If they are above 16, print “You can drive”
- TODO four asks the user to print “you can drive” if the user is above 16 but below 18. It should print “You can drive and vote” if the user is above 18 but below 21. If the user is above 21, it should print “you can drive, vote and drink (but not at the same time!”.
- On a separate screen, there is a number input box and a ‘sum’ text label and an ‘add’ button. TODO five asks the user to display the cumulative sum of all numbers added every time the ‘add’ button is pressed.

- TODO six is on a separate screen with a number input text field and a button, ‘is even?’. When the button is pressed, a message should be printed indicating whether the number is even.
- TODO seven is on a separate screen with yet another text input box and a ‘calculate’ button. It prompts for the user to create a ‘fibonacci adder’ class with a method ‘fibonacciNumberAtIndex(indexOfFibonacciNumber: int)’. That class must be hooked up to the screen, and print out the appropriate fibonacci number of an inputted integer.

4) Deliverables

- Assignment (code, sandbox file) posted on Github

5) Timeline:

- Due on first day of week 3

6) Suggested ways to get started

- Answer the following questions:
  - What’s a method? What’s a variable?
  - How do you make a hook into your code from something you have defined in Interface builder?
  - When might you use a for loop? A while loop?
  - What is a function? When might you use one?
  - What is a variable, and how is it different from a constant?
  - What is a conditional? A boolean?
  - What are types? Is swift ‘typed’?

7) Resources

- Apple’s guide to Swift control flow: https://developer.apple.com/library/prerelease/mac/documentation/Swift/Conceptual/Swift_Programming_Language/ControlFlow.html
- Apple’s guide to writing functions in Swift: https://developer.apple.com/library/prerelease/mac/documentation/Swift/Conceptual/Swift_Programming_Language/Functions.html#//apple_ref/doc/uid/TP40014097-CH10-XID_245

8) Evaluation

Your assignment will be evaluated regarding the extent to which you meet the above requirements using this rubric.

The rubric outlines how your assignment will be evaluated on assignment readiness, stability & performance, and style & readability.