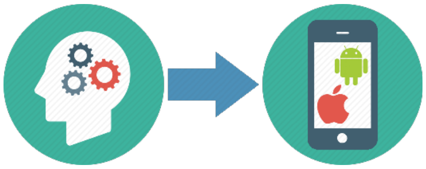
Mobile App Development

CS 4153



Weekly Assignment 01

Last updated: Aug 23, 2018 12:30 PM Due: Aug 30 11:59PM

This assignment will give you an opportunity to try out the Swift programming language using a project and a playground within Xcode 10 beta.

1. Create an Xcode 10 beta project.
   - Use the *Cocoa Touch Framework* template.
   - Specify the product name as:

     W01_ lastName_firstName

     where *lastName* is the part of your name that OSU considers to be your last name, and *firstName* is your first name. (This is the product naming convention we will use for all weekly assignments.)

2. Add a new Swift file named *Cash.swift* to your project.

3. Define a class public named *Cash* within *Cash.swift.* The class contains the following public elements:
   - A `Double` stored property that contains the amount of money (dollars and cents) described by an object of the class.
   - A read-only, computed property. The *getter* calculates and returns the minimum number of U.S. bills and coins that add up to the amount in the stored property. The return value is an `Int` array of length 9 that contains (beginning with index 0 of the array) the number of $50 bills, $20 bills, $10 bills, $5 bills, $1 bills, 25¢ coins, 10¢ coins, 5¢ coins, and 1¢ coins. For example, if the stored property contains 47.23, the return value is:

     [0, 2, 0, 1, 2, 0, 2, 0, 3]

     However, if the amount in the stored property is negative, return *nil*.
   - An initializer with one `Double` parameter; it assigns the parametric value to the stored property.

4. Add a Swift playground to your project. Name the playground *MyPlayground.*

5. In the playground:
   - Import your framework.
   - Define 6 variables of type *Cash*. Initialize one to a negative number; initialize another to zero; initialize the remaining variables to "random-ish" amounts in the range of 0 < amount <= 100.
   - Print the values of the variables to the debug area.

Xcode caution!

If you have built your framework after any changes and get a *No Such Module* error when importing your framework, do the following. (It works for me, though it makes no sense.)

1. Click on the project item (the one with the blue "A" on it) in the project navigator.
2. Now click on the playground item in the project navigator. The error should disappear after a few seconds.

General notes

- Since this is a senior/graduate course, you are expected to use good development practices for all programming assignments. For example: comment your code, use blank lines to separate functionality, and use meaningful variable names.
  - If such practices are not followed, your score may be reduced.
- You may search the Web or a book for an algorithm that converts cash amounts into a minimal number of bills and coins. If you use such a source, make sure to include a comment box at the beginning of your *getter* in which you provide a reference to the source from which you got the code/idea.
  - Proper referencing will be expected throughout the semester.
- Points will be deducted for extraneous semicolons and for extraneous parentheses in flow control statements (loop and decision statements).

Submitting your solution

- To compress a file or folder on macOS, right-click on the file or folder name, and then select *Compress …* from the context menu that appears. A *zip* file with the same prefix as the file or folder will appear.
- After closing your project in Xcode, compress (*zip*) the project folder and submit it to the appropriate dropbox on the course BrightSpace page.