

Homework #1– Swift 101

CIS380 –Summer 2015

Due Date: 8am, July 14, 2015

Learning Objectives

- Get familiar with Swift syntax.

Instructions

Please work in pairs on the following 20 coding exercises.

Description

Referencing the material covered in lecture on Swift as well as the reading assignment, create a new Playground in xCode and solve the following programming exercises in Swift. Please include all of your solutions in a single Playground and use comments to associate each solution to the problem listed below as follows:

```
// Problem #1
let myString = "hello"
.
.
.
```

Problem 1: Define the following variables/constants:

- a constant myString that has implicit type String and value “hello”.
- a variable cost that has implicit type Double and value 3.14.
- a constant cnt that has explicit type Int and value 2.
- a variable shouldWe that is typed as a boolean value and set to true.
- an integer constant assigned the value of 11 hexadecimal.
- an integer constant assigned the binary equivalent of decimal 10.

Problem 2: Use the \() operator to form a string from string literal data and a floating point calculation.

Problem 3: Use arrays in Swift to accomplish the following:

- Define an array that consists of 3 string values: “queen”, “worker” and “drone”.
- Use the println function to printout the first entry in the array.
- Use the append() method to add an addition entry of “honey” to the array.
- Use the += operator to add the [“are”, “us”] to the array.

Problem 4: Write two different style for loops (e.g. for item in, and for (index,value) in enumerate()) that iterate through the array created in problem 3 and prints the values. In the first example print only the value, for the second each index should be printed with the value. e.g. “Item #0 is queen” ...

Problem 5: Create a variable that is explicitly typed as a dictionary that maps strings to floating point numbers. Initialize the variable to the data shown in the table below which lists an author name and their comprehensibility score.

“Mark Twain”	8.9
“Nathaniel Hawthorne”	5.1
“John Steinbeck”	2.3
“C.S. Lewis”	9.9
“Jon Krakaur”	6.1

Problem 6: Using the dictionary created in the previous problem, do the following:

- Print out the floating-point score for “John Steinbeck”.
- Add an additional author named “Erik Larson” with an assigned score of 9.2.
- Write an if/else statement that compares the score of John Krakaur with Mark Twain. Print out the name of the author with the highest score.

Problem 7: Use a for loop to iterate through the dictionary created in problem #5 and print out the content in the form of key: value, one entry per line.

Problem 8: Write a for loop where the index variable ranges from 1 to 10. Print the index variable during each iteration.

Problem 9: Write a “C style” for loop where the index variable ranges from 10 down to 1. Print the index variable during each iteration.

Problem 10: Underscore notation can be used to craft a for loop where the index variable is not needed. Use this notation to write a for loop that computes the product of $x * y$ using only the $+$ operator.

Problem 11: Use a while loop that computes the average of the scores in the dictionary created in problem #5.

Problem 12: Based on the outcome of problem #11, write an if/else statement that prints out “Low” if the average is less than 5.0, “Moderate” if the average is ≥ 5 but less than 7, or “High” if the average is ≥ 7 .

Problem 13: Write a switch statement that examines a variable named count and assigns a string variable named strOut the following values.

0	“none”
1 - 3	“a few”
4 - 9	“several”
10 - 99	“tens of”
100 - 999	“hundreds of”
1,000-999,999	“thousands of”
> 999,999	“millions of”

Problem 14: Write a Swift function named “verbalizeNumber” that takes an input parameter of type Int, and returns a String. The function should utilize the switch statement written in the previous exercise to map the integer value to a verbal description.

Problem 15: Write a C-style for loop that iterates from 1 to 100_000_000 where your index is updated on iteration by multiplying by 10, e.g. index *= 10. In the body of the loop print out a meaningful string that is in part generated by calling the verbalizeNumber function written in problem #14.

Problem 16: Write another Swift function named verbalizeAndShoutNumber that is identical to the one you wrote in problem 14, only all letters in the expression returned are capitalized. Note, if you research how you can capitalize a string in Swift, you should be able to implement this function in a single line of code!

Problem 17: Wrap the functionality you wrote in Problem 15 as a function named expressNumbersElegantly that takes two parameters – an Int value, and a function that takes an Int and returns a String. The implementation of this function should do exactly what you did in 15 only:

- iterate up to the integer value passed in.
- call the function passed as a parameter instead of hardwiring the call to verbalizeNumber.
- instead of printing out text, you should concatenate it into a single string and return it to the caller.
- Define a variable of type function that takes an Int value and returns a String. Set the variable to each of the two functions (verbalizeNumber and

verbalizeAndShoutNumber) and call `expressNumbersElegantly` each time you set it.

```
func expressNumbersElegantly(max: Int,  
    verbalizeFunction: (Int) -> String) -> String
```

Problem 18. Rewrite the `expressNumbersElegantly` function in the form of a new function named `expressNumbersVeryElegantly` that uses external parameter names. Be sure to come up with nice external parameter names so the invocations read very elegantly!

Problem 19: Given the following Swift array:

```
var famousLastWords = ["the cow jumped over the moon.", "three score  
and four years ago", "lets nuc 'em Joe!", "ah, there is just something  
about Swift"]
```

use the `map` function `Array` class to capitalize the first letter of each entry in the array. Code up the capitalization code as a closure that is passed to the `map` function.

Problem 20: Write a short clever and concise fragment of Swift to demonstrate your newly acquired Swiftian prose to your very proud instructor!

Deliverables

To receive credit for your homework, you must:

- Hand in a hardcopy of printout of your Playground containing solutions to the above problems. Make sure both partners' names are in the header to get full credit!
- Create a zip archive of your entire project folder and name it `Lab01-TeamDD.zip` where `DD` is replaced with your team number. Upload your zip archive to DropBox or equivalent and email a public link to the instructor (jonathan.engelsma@gvsu.edu) with the email subject set to `Lab01-TeamDD`. Failure to follow naming instructions correctly means your homework will not get graded and you will get a zero!