CM 161 In Class Assignment for January 29, 2018
Swift Playgrounds #3: Introduction to Arrays
Points Possible: 20 points
Due on: Tuesday, 2/13/18 in your Github

Part 1: Getting set up in Github

1. Create a new repo in your Github and name it "**SwiftPlaygroundThree**"
2. Clone the repo to your desktop
3. In Xcode, create a new playground and name it, "**SwiftPlaygroundThree**"
4. Quit Xcode
5. Open the local repository directory (folder) and drag and drop in your **SwiftPlaygroundThree**" playground file
6. Do an initial commit and push to Github
7. Open up your playground file in your directory and follow instructions below

Part 2: Introduction to Arrays

1. What is an array? Arrays are one of the most commonly used data types in an app. You use arrays to organize your app's data. Specifically, you use the Array type to hold elements of a single type, the array's Element type. An array can store any kind of elements—from integers to strings to classes.

Swift makes it easy to create arrays in your code using an array literal: simply surround a comma-separated list of values with square brackets. Without any other information, Swift creates an array that includes the specified values, automatically inferring the array's Elementtype. (reference: https://developer.apple.com/documentation/swift/array)
In other words, we can think of arrays as a list of items.

Q: What are some data types that we've been using so far in this class?

Q: Can you guys guess how to create a variable that is an array?

Part 2: Declare an array of type String

1. Let's go ahead and create an array of type String that holds a list of fruit names.

Declaration: var fruitNames : [String]

We've just declared an array that contains strings. The [ ] brackets here indicate that we are declaring an array.

Q: Do you remember how to declare a variable of type Int?

Part 3: Initialize an array using an Array Literal
1. Using an Array Literal, we're going to go ahead and add some fruit names to fruitNames.

fruitNames = ["Kiwi", "Apples", "Watermelon"]

Q: So, now we've got a few items in our array. Why is it necessary that we put "" marks around each of the items in fruitNames?

Part 4: On Your Own: Print the Array
1. Use the print function to print, fruitNames.

2. Ok, if you had trouble figuring it out, remember that if you want to print a variable, we can simply use the **print()**. The item that we want to print has to go in between the parenthesis.

print(fruitNames)

Part 5: So, what happens if we want to use string interpolation and add the string, "I like to eat" to each item of the array and print it out? In other words, we want the following to print in the console in this way:

"I like to eat Kiwi"
"I like to eat Apples"
"I like to eat Watermelon"

First try this. This is what we're familiar with when using string interpolation with a string variable.

print("I like to eat \(fruitNames)")

So, this is what prints out in the console, and it's not exactly what we're looking for.

Part 6: We're going to have to iterate through the array using the for-in syntax

1. Ok, what does this mean? We have an array and the for-in allows us to iterate through the array by performing a task over each item in the array until all items in the array have gone through the iteration.

Let's go ahead and right up some code:

```
for name in fruitname {
print("I like to eat" + name)
}
```

It's important to note that "name" can be replaced by any word here, but it holds the data temporarily during the iteration so that it can be printed.

Part 7: Introduction to array indexes

1. We are able to print out an item from an array based on its index, or the location it sits in the array. Let's take a look at the index of each item in our current array, fruitNames.

```
fruitNames = ["Kiwi", "Apples", "Watermelon"]
```

It's important to note here that indexes start at zero.

Q: What index is Apples at? A: Index 1

2. How do we print out a particular item at a specific index in an array? For example, how do we access and print only "Apple"?

```
print(fruitNames[1])
```

3. On your own: print out the item at index 0.

```
print(fruitNames[0])
```

Review:

1. We can think of arrays as a list that can hold items of specific data types
2. We can add items to an array using the **Array Literal**
3. We can iterate through an array and perform tasks on each item in the array using the **for-in** syntax
4. We can access a particular item in the array via it's **index**.