

iOS Application Development Swift Assignment

Assignment 4

Working with Closure:

Problem 1: Defining closure variables and constant

Define the following:

- A variable called func1 whose type is (Int, Int) -> Int
- A variable called func2 whose type is (String) -> Void
- A constant called func3 whose type is () -> Int
- A constant called func4 whose type is () -> [() -> Void]
- A constant called func5 whose type is [(int) -> Void] -> [String]
- A variable called func6 whose type is [(int) -> Void] -> [(int) -> void]
- A variable called func7 whose type is [(int) -> Void]? -> [(int) -> void]?
- A variable called func8 whose type is [(int) -> Void]? -> [(int) -> void]?
- A variable called func9 whose type is () -> ((Int) -> Int, Int)?

Problem 2: Dummy functions

For each of the variable/constant above define a dummy function and assign the corresponding variables or constants to them. For instance:

- Variable called func0 whose type is String -> Void
- var func0: (String) -> Void
func fakeFuntion0 (s: String) -> Void {
 print(s)
}
func0 = fakeFuntion0;

Problem 3: A Calculator

- Define an array whose type is String

- The size of the array is between 5-10 (pick an arbitrary number)
- Initialize the array with some initial values.
 - The possible values the items of the array can take is following
 - sum
 - division
 - difference
 - power
 - multiplication
- Define a function for each of the above mathematical operations (sum, division, difference, complement, power, multiplication)
- Define a dictionary as following
 - The type of key is String
 - The type of the value is a function type (Int, Int) -> Double
- Define a function called calculator which takes the following as inputs:
 - Two numbers
 - An operation (which is either sum, division, difference, power or multiplication)
- The calculator function returns a tuple with two parameters: 1- the result of the operation and 2- the operation's corresponding function. For instance, of the operation sent to the function is sum, and the two numbers are 10 and 20 the function will return a tuple which is (20, the sum function from the dictionary).
- Write a for-loop and iterate over the items of the array of the operations and call the calculator function for each item of the array.

Problem 4: Sorted()

Write a function called sorter which receives two inputs parameters:

1. A list of String called words
2. A function type of type (String,String)->Bool called comparator

And returns a list of String

The function sorter, then will sort the list (array) of String using the criteria defined by the comparator parameter.

```
func sorter(_ list:[String]?, criteria by: (String, String)->Bool) -> [String]? {
  guard let words = list else {
    return nil
  }
  sortedWords = words.sorter(criteria)
  return sortedWords
}
```

Now call the sorter function defined above for the following list and criteria:

1. List: ["one", "two", "three", "four", "five"]
Criteria: Alphabetically Ascendingly
 2. List: ["one", "two", "three", "four", "five"]
Criteria: Alphabetically Descending
 3. List: ["aa", "aba", "b", "aabbb"]
Criteria: Descending based on the length of the items
-]

Problem 5: Query Builder

Write a function called filter which has two input parameters:

1. A list of numbers (positive integers)
2. An array of predicates. (A predicate is a function type of type (Int)->Bool).

And returns an Optional Array of Integer.

Then define the following constant predicate and add them to a list of predicates.

Predicate1: Returns true if a number is odd

Predicate2: Returns true if a number is prime

Predicate3: Returns true if a number is prime and odd

Predicate4: Returns true if a number is divisible by 7

Add the above predicates to a list of predicates.

Then implement the filter function mentioned above. The filter function returns the list of all numbers from the input list on which all predicates return true.