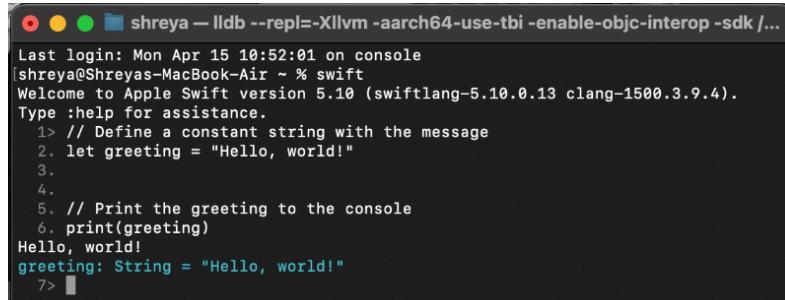


SWIFT Code - Shreya & Christina

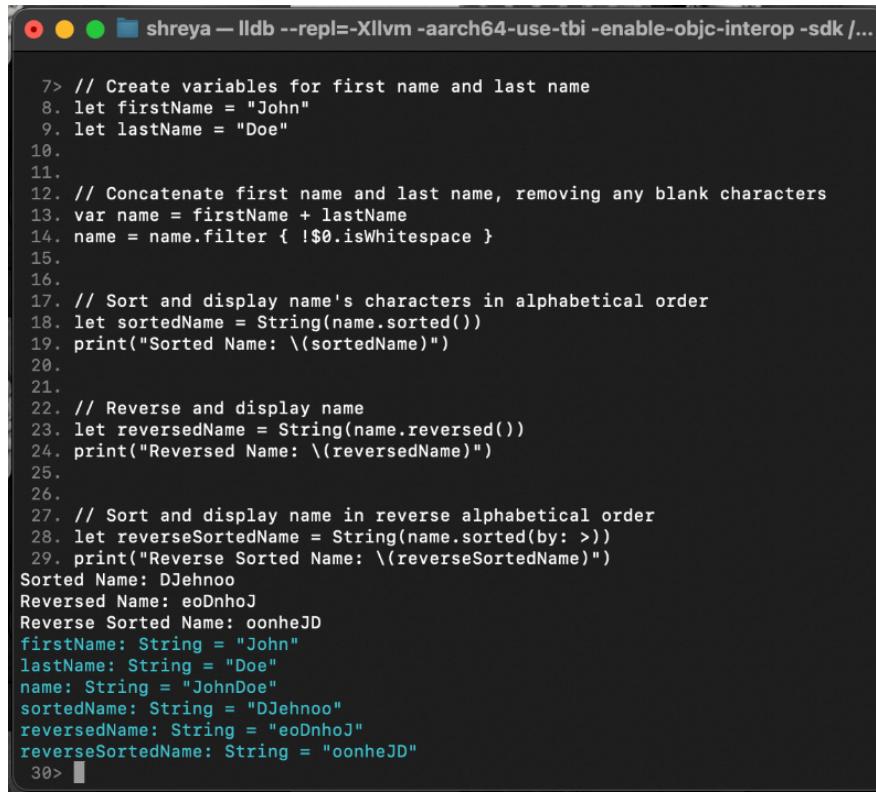
Step 1 – Install Swift to run a local interactive REPL shell or use an online REPL as discussed in class and perform a “Hello World!” program.



```
shreya — lldb --repl=-Xllvm -aarch64-use-tbi -enable-objc-interop -sdk /...  
Last login: Mon Apr 15 10:52:01 on console  
[shreya@Shreyas-MacBook-Air ~ % swift  
Welcome to Apple Swift version 5.10 (swiftlang-5.10.0.13 clang-1500.3.9.4).  
Type :help for assistance.  
1> // Define a constant string with the message  
2. let greeting = "Hello, world!"  
3.  
4.  
5. // Print the greeting to the console  
6. print(greeting)  
Hello, world!  
greeting: String = "Hello, world!"  
7> █
```

Step 2 – Install Swift to run a local interactive REPL shell or use an online REPL as discussed in class and perform the following

- Create a string assigned to the variable FirstName
- Create a string assigned to the variable LastName
- Create a string assigned to the variable Name that concatenates FirstName and LastName, removing any blank characters
- Sort and display Name's characters in alphabetical order
- Reverse and display Name o Sort and display Name in reverse alphabetical order
- Include comment lines in your REPL for citations of helpful or copied sources
- Capture all the REPL output into a Step1.txt file



```
shreya — lldb --repl=-Xllvm -aarch64-use-tbi -enable-objc-interop -sdk /...  
  
7> // Create variables for first name and last name  
8. let firstName = "John"  
9. let lastName = "Doe"  
10.  
11.  
12. // Concatenate first name and last name, removing any blank characters  
13. var name = firstName + lastName  
14. name = name.filter { !$0.isWhitespace }  
15.  
16.  
17. // Sort and display name's characters in alphabetical order  
18. let sortedName = String(name.sorted())  
19. print("Sorted Name: \(sortedName)")  
20.  
21.  
22. // Reverse and display name  
23. let reversedName = String(name.reversed())  
24. print("Reversed Name: \(reversedName)")  
25.  
26.  
27. // Sort and display name in reverse alphabetical order  
28. let reverseSortedName = String(name.sorted(by: >))  
29. print("Reverse Sorted Name: \(reverseSortedName)")  
  
Sorted Name: DJehnoo  
Reversed Name: eoDnhoJ  
Reverse Sorted Name: oonheJD  
firstName: String = "John"  
lastName: String = "Doe"  
name: String = "JohnDoe"  
sortedName: String = "DJehnoo"  
reversedName: String = "eoDnhoJ"  
reverseSortedName: String = "oonheJD"  
30> █
```

Step 3 – Write a Swift program that prints out a Six Little Words puzzle.

Create a standalone Swift program that will successfully run if called using swift. The code should instantiate an object from a class called SixLittle and call that object's Run method.

Create a SixLittle class in Swift that has the following four methods:

Ask – asks for six pairs of string inputs – a word of at least 4 characters length (should re-ask if too short) followed by a related hint string (of any length) – this pair of words should be kept in collection attributes (words, hints) inside the SixLittle class.

Prepare – The prepare method will break each word in the words collection in half (be consistent about what to do for an odd number of letters), the broken parts of the words will be capitalized and then added to a collection called tokens. Finally, the tokens collection should be randomly shuffled so that the tokens are not in the order they were originally placed.

Display – displays the title “Six Little Words (Swift)”, displays a neat table titled “Tokens” of the tokens – 4 words across in 3 rows, displays the hints in a list titled “HINTS”, and then display another list labeled “ANSWER KEY” and list the original words on two lines, separating each section with a blank line.

Run – should clear all collections, call Ask, Prepare, and Display in turn.

Typical output when program is run and six words/hints have been entered by the user:

Six Little Words (Swift)

Partial Words:

LA ACH KE

AKE CH BE

AIR DO IR

FA SN OR

Hints:

Open me

I slither

A carnival

Sit on this

Fresh water body

Sand and sun

Answer Key:

DOOR SNAKE FAIR

CHAIR LAKE BEACH

```
1 import Foundation
2
3 class SixLittle {
4     var words: [String] = []
5     var hints: [String] = []
6     var tokens: [String] = []
7
8     func ask() {
9         print("Please enter six pairs of words and related hints:")
10        for _ in 0...<6 {
11            var word = ""
12            var hint = ""
13            repeat {
14                print("Enter a word of at least 4 characters:")
15                if let inputWord = readLine(), inputWord.count >= 4 {
16                    word = inputWord
17                } else {
18                    print("Word must be at least 4 characters long. Try again.")
19                }
20            } while word.isEmpty
21
22            print("Enter a related hint:")
23            if let inputHint = readLine() {
24                hint = inputHint
25            }
26
27            words.append(word)
28            hints.append(hint)
29        }
30    }
31 }
```

```
31
32     func prepare() {
33         tokens.removeAll()
34         for word in words {
35             let splitIndex = word.count / 2
36             let firstHalf = word.prefix(splitIndex).capitalized
37             let secondHalf = word.suffix(word.count - splitIndex).capitalized
38             tokens.append(firstHalf + " " + secondHalf)
39         }
40         tokens.shuffle()
41     }
42
43     func display() {
44         print("\nSix Little Words (Swift)\n")
45         print("Partial Words:")
46         printTable(tokens, columns: 4, rows: 3)
47         print("\nHints:")
48         for hint in hints {
49             print("- \(hint)")
50         }
51         print("\nAnswer Key:")
52         for (index, word) in words.enumerated() {
53             if index % 2 == 0 && index != 0 {
54                 print()
55             }
56             print(word, terminator: " ")
57         }
58         print()
59     }
60 }
```

The screenshot shows the Xcode interface with the project 'Capstone' open. The main window displays the 'main.swift' file. The code defines a 'SixLittle' class with a 'run()' method that calls 'ask()', 'prepare()', and 'display()'. It also includes a 'printTable()' function for printing arrays and an extension for 'Array' that provides a 'chunked(into:)' method. A comment indicates the instantiation of a 'SixLittle' object and the execution of its 'run()' method.

```
func run() {
    ask()
    prepare()
    display()
}

func printTable(_ array: [String], columns: Int, rows: Int) {
    let chunkedArray = array.chunked(into: columns)
    for chunk in chunkedArray {
        for item in chunk {
            print(item, terminator: " ")
        }
        print()
    }
}

extension Array {
    func chunked(into size: Int) -> [[Element]] {
        return stride(from: 0, to: count, by: size).map {
            Array(self[$0 ..< Swift.min($0 + size, count)])
        }
    }
}

// Instantiate SixLittle object and run the program
let game = SixLittle()
game.run()
```

The screenshot shows the terminal window with the 'Run' tab selected. The application interacts with the user to find words based on hints:

- Enter a related hint: Open me
- Enter a word of at least 4 characters: SNAKE
- Enter a related hint: I slither
- Enter a word of at least 4 characters: FAIR
- Enter a related hint: A carnival
- Enter a word of at least 4 characters: CHAIR
- Enter a related hint: Sit on this
- Enter a word of at least 4 characters: LAKE
- Enter a related hint: Fresh Water Body
- Enter a word of at least 4 characters: BEACH
- Enter a related hint: Sand and sun

Six Little Words (Swift)

Partial Words:

Be Ach La Ke Do Or Sn Ake
Ch Air Fa Ir

Hints:

- Open me
- I slither
- A carnival
- Sit on this
- Fresh Water Body
- Sand and sun

Answer Key:

DOOR SNAKE

FAIR CHAIR

LAKE BEACH