**INFO 6350 Fall 2022**

**Assignment 3**

Using Swift playground and / or the command line for macOS (open XCode, create a new XCode project, macOS, command line tool), practice the following exercises:

**Exercise: Functions**

1. Create a function that implements the factorial operator( ! )
2. Create a function that takes an array of integers as input and split it into two arrays

of odd and even integers. Print the result on the console(two arrays).

1. Create a function that takes a string as input and checks whether the string is

palindrome or not. Print the result on the console (true or false).

**Exercise: Structures**

1. Create a structure **Student.** This structure should have

• four properties: name, age, id, and GPA.

• a function that can change the name.  
• a function that can change the GPA.

2. Create a student instance and call these functions.  
3. Print the new value of all the four properties on the console

**Exercise: Classes**

1. Create a class **Ipad.** This class should have

• an empty function *doesSupportApplePencil()*.

• an empty function *sizeOfScreen()*;

1. Create a classe (for example IpadPro5) as a subclass of **Ipad** and override these functions.
2. Call these functions.

**Exercise: Optionals**

1- Ask the user to enter Mobile Number, and take input using the command line.

• If no input was provided print ‘No Mobile number provided’  
• If the input is all numbers print ‘Mobile Number in Digits’ followed by the number • Otherwise, print ‘Mobile Number in String’ followed by the String entered

**Exercise: Protocols**

1. Create a protocol **TransportationTool.** This protocol should have

* a function that returns the number of wheels
* a function that returns the way of transportation (in the air, on the ground, on the water).

• a function that takes place of departure and destination as inputs and prints a sentence like “From xx to xx by xx”.

2. Define two classes that conform to this protocol. (car, ship, bicycle, etc.)

**Exercise: Closures**

1. Create a closure that takes two Integers as inputs and returns the sum.

**Exercise: Enumerations**

1. Rewrite the way of transportation of **TransportationTool** into an enumeration.

2. Create an associate enumeration that has:

• a case **GPA** that associates two Double values(like 3.9, 4.0, 2.6).

• a case **Grade** that associates two String values(like “A”,”B”,”C”).

3- Create a **GPA** enumeration instance and **Grade** enumeration instance

**Exercise: Extensions**

1. Create a class **Distance** that has a property: *valueInKilometer*.

2- Create an extension to the **Distance** class that has:

* a function that returns the mile value of *valueInKilometer.*