INFO 6350 Spring 2019

Assignment # III

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

*Exercise: Swift Classes*

1. Define a class **Employee** having 3 properties: name, age and gender.
2. Define a custom initializer in the Employee class to set only the name of the Employee instance.
3. Create another class named **Clerk** and assign **Employee** as the parent/superclass of the Student class.
4. Initialize a Manager instance by invoking the custom initializer defined in the Employee class.
5. Instantiate an Employee instance “**emp1**”, set its properties and assign **emp1** to **emp2**.
6. Update the name property of **emp1**.
7. Print the value of the name property of both Employee instances.

*Exercise: Enum*

1. Define an enum **Day** which consist of all the days of the week.
2. Next, define a function **dayType()** which accepts the day and prints out “Weekend” for “Saturday” or “Sunday” and “Weekday” for the remaining days. **{Use switch case statement}**

*Exercise: Structures*

1. Define a structure, **Point**, having 2 parameters x and y, to encapsulate the data to store a coordinate of a point (x,y) in a two-dimensional space.
2. Instantiate **point1** with values of x and y co-ordinates as 0.
3. Assign point1 to **point2**.
4. Set the x co-ordinate of point1 to a different value.
5. Print the x co-ordinates of both the points.

*Exercise: Protocols*

1. Define a protocol **Animal**, inside this protocol define a function(s)
2. **Legs**() – This should print the number of legs
3. **Hands()** – This should print the number of hands
4. Define a class **Kangaroo** which adopt to Animal and provide the actual implementation.

*Exercise: Extensions*

1. Create a *calculator* functionality by extending the datatype**Double**

* Add methods for Addition, Subtraction, Multiplication, Division, Modulus. {Use the second number as 100}

1. Next, create a constant named **cal1** of **Double**initializing it to a number of your choice. Use the extended functionality in point 1 and print out the calculated values in the console.
2. Complete the code below:

struct Circle{

var radius : Double 

***//Define an initializer***

}

extension Circle { 

***//Define a method to print out the area of a circle using the radius***

}

var c = Circle(radius: 3)

c.areaOfCircle()