Assignment 7

Problem0

Write and run all the sample codes in the Protocol chapter and then do the following problems:

Problem1

Consider the following protocol:

```
Protocol SomeProtocol1
{
        var property1: Int {get}
        var property2: String {get set}
        var property3: (()->Int)? {get}
        static var property4: String? {get set}
        static method1(arg param: Int)
        method2(arg param: Int) -> Int?
        mutating method3()-> String?
}
```

Now

- Create a class called Class1 which confirms to the above protocol. Make some arbitrary implementations for the protocol's requirements if needed.
- Create a struct called Struct1 which confirms to the above protocol. Make some arbitrary implementations for the protocol's requirements if needed.
- Create an enum called Enum1 which confirms to the above protocol. Make some arbitrary implementations for the protocol's requirements if needed.

Problem2

1. Create a Swift playground. Create a swift class called CarRepairServiceProvider which provides a car repair service. This class requires some information to work: a) Description issue and b) car model.
2. Create another service provider class called, HomeServiceProvider which performs three (3) types of home services 1) CookingService 2) WashingService 3) BabySittingService. Create an enum called HomeService with the above items. This class also requires some information to work: 1- service type (any of the above services) and 2- address the service should be done in.
3. Create a new service provider Swift class called, StudentServicesProvider which provide one student service. It needs the following requirements: 1- student service description an
4. Define a clientA class who is interested in CarRepairServiceProvider.
5. Define a clientB class who is interested in HomeServiceProvider.

6. Define a clientC class who is interested in StudentServicesProvider.
7. Define a clientD class who is interested in all above services.
8. Use delegation pattern to implement this problem.

Problem3

In Problem 1 you are going to improve the code you have done in Problem0.

1. Create a Service Provider class as a parent (base) class and make all your service classes (CarRepairServiceProvider, HomeServiceProvider, StudentServicesProvider,…) inherits from this parent class. Make any other necessary changes.
2. Create an protocol which is the parent of all protocols you have defined in your Problem0. Now think about this: Now, do you need the other protocols at all anymore? If No, then remove them and make corresponding changes. If Yes, then make corresponding changes.

After the making any possible changes your application should still be able to deliver the same functionalities as Problem 0.

Problem4

Imagine you design a mobile view or a webpage which contains a table (Similar to the picture below). The table lists the students in our program. Create an application using the delegate approach which given a list of students it create the table. At the minimum you need two classes View and Table.

In order to create the table, the Table should know about the following information. Also Imagine the View has a list of students and it uses a Table object to show the list of Students.

- The View class has the list of students.
- The list of students is not fixed and could be arbitrary. In other word, if there are 10 students in the list, then the table has 10 rows. If there are only 2 students in the list, then table shows 2 rows and so on.
-

Ali David, 80
Peter Jackson, 79
Donald Trump, 81
William Chen, 76
Ana Leu, 82

- The Table class is responsible for showing the information in the table.
- The Table requires the following information to be able to show the information
-
    - Student data. Each row represents one students
    - How many students are in the list
    - The fixed height of each row

- As mentioned, each row in the table represents an item which contains the following properties 1- firstName 2-LastName 3- GPA

Problem5

Consider the following protocol:

```
protocol SpecialStringProtocol
{
  //it checks whether a given condition is met on the current string
        func verifiableStringForACondition( _  conditionPredicate: String->Bool) -> Bool

     /*it checks whether the string is a double repeated string. For instance abab is a double
repeated because the ab is repeated 2 times. Or AliAli is also double repeated.*/
        func repeatedString()-> Bool
}
```

Now, extend the String type to have the above requirements defined in the above protocol and also write some test scenarios to show how it works.

Problem6

Consider the following protocols:

```
Protocol Protocol1 {
        var property1: Int?
}

Protocol Protocol2 {
        var property2: String
}

Protocol Protocol3 {
        var property3: Int?->String
}
```

- Define an array called list1 whose elements are of a type (for instance a class) which conforms to all above protocol.
- Using protocol composition, define a dictionary whose keys are of type String and the values are of a type that conforms to all above (three above) protocols.