軟體人才培訓學院 / **iOS Class**

Remote Learning Assignment - Week 2

---

**Part 1: Object-Oriented Swift**

1. Declare a class `Animal` with a property `gender` and a method `eat()`. The data type of `gender` should be enum Gender as below and when you call `eat()` method, it will print “I eat everything!”

```
enum Gender {
    case male
    case female
    case undefined
}
```

2. Declare three classes: `Elephant`, `Tiger`, `Horse` that inherits from Animal and override the `eat()` method to print what they usually eat.

3. Declare a class `Zoo` with a property `weeklyHot` which means the most popular one in the zoo this week. The codes below can’t work correctly, please find what data type should `A` be and solve the problem. Note that `tiger`, `elephant`, and `horse` are instances of class Tiger, Elephant, and Horse respectively.

```
class Zoo {

    var weeklyHot: A

    init(weeklyHot: A) { }
}

let zoo = Zoo(weeklyHot: Tiger())

zoo.weeklyHot = tiger
zoo.weeklyHot = elephant
zoo.weeklyHot = horse
```

4. What is an instance? What does `Initilizer` do in Class and Struct?

5. What’s the difference between `Struct` and `Class`?

6. What’s the difference between `reference type` and `value type`?

7. What’s the difference between `instance method` and `type method`?

8. What does `self` mean in an instance method and a type method respectively?

---

**Part 2: Enumerations and Optionals in Swift**

1. There are several gasoline types, 92, 95, 98, and diesel that we can use enum to model them.

    - Please declare an enum named `Gasoline` to model gasoline.
    - Every kind of gasoline has its price. Please declare a computed property named `price` and a method named `getPrice` separately in Gasoline enum that both will return different prices depending on different gasoline.
    - Please establish `raw values` for `Gasoline`. The data type of raw value should be String. For example, `Gasoline.oil92.rawValue` should be “92”.
    - Please explain what enum `associated value` is and how it works.

2. Optional is a very special data type in Swift. Take `var a: Int? = 10` for example, the value of `a` will be `nil` or `Int`. You should have learned how to deal with Optional.

    - People would like to have pets, but not everyone could have one. Declare a class `Pet` with `name` property and a class `People` with `pet` property which will store a Pet instance or nil. Please try to figure out what data type is suitable for these properties in Pet and People.
    - Please create a People instance. Use `guard let` to unwrap the `pet` property and print its name.
    - Please create another People instance. Use `if let` to unwrap the `pet` property and print its name.

### Part 3: Protocol in Swift

1. Declare a struct `Person` with a `name` property type String and a protocol name `PoliceMan`. There is only one method `arrestCriminals` with no argument in the protocol.

2. Make struct `Person` conform to `PoliceMan` protocol.

3. Declare a protocol `ToolMan` with a method `fixComputer` that has no argument.

4. Add a property `toolMan` to the struct Person with data type `ToolMan`.

5. Declare a struct named `Engineer` that conforms to the `ToolMan` protocol.

6. Create a Person instance with the name “Steven” and also create the relative data you need to declare this instance.

### Part 4: Error Handling in Swift

```
enum GuessNumberGameError {
    case wrongNumber
}

class GuessNumberGame {

    var targetNumber = 10

    func guess(number: Int) throws {
        guard number == targetNumber else {
            throw GuessNumberGameError.wrongNumber
        }
        print("Guess the right number: \(targetNumber)")
    }
}
```

Read the code above first and paste it in the playground file, there is an error inside the code. Please solve the error by adding a piece of code in the file. Call `guess(number:)` and pass 20 as the argument after you fix the problem.

## Part 5: A Simple App

Please implement the app by following the design below (measured in points.)

### UI Requirements:

Label on the top:
Top: 100, Leading: 40, Font size: 16, Font Color: White

Label in the middle:
Top: 100, Leading: 40, Trailing: 40, Font size: 16, Font Color: White

Button:
Bottom: 50, Leading: 40, Trailing: 40, Font size: 16, Font Color: White


### Functional Requirements:

Each time the user hits the button, the background color and text should change randomly. We give the text below. There are 7 sentences in the array. You can choose 7 kinds of color to cooperate with it. For example, you can take the colors of the rainbow as your color set.

**Hints:**

1. RGB color
2. Random number in Swift

```
let text = [
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas tempus.",
    "Contrary to popular belief, Lorem Ipsum is not simply random text.",
    "Richard McClintock, a Latin professor at Hampden-Sydney College in ",
    "looked up one of the more obscure Latin words, consectetur",
    "from a Lorem Ipsum passage, and going through the cities of the word",
    "This book is a treatise on the theory of ethics, very popular during the.",
    "The first line of Lorem Ipsum, Lorem ipsum dolor sit amet..",
]
```