Structures and Classes

By: Zander Ewell

What is a Struct (Structure)?

A structure is a group of one or more properties that make up a type. It is used to store variables of different data types.

Structure example

```
import UIKit

struct Dog {
    var name: String
    var color: String


func dogTalks() -> String{
    let doggo: String = "Hello I am a dog. MY name is \(name) I am \(color) and fluffy! Would you like to pet me?"
return doggo
    }
}

let dog = Dog(name: "joe", color: "White")
print(dog.name)
print(dog.color)

print(dog.dogTalks())
```

Output:

Joe

White

Hello I am a dog. My name is joe I am White and fluffy! Would you like to pet me?

What is a class?

Classes are similar to structures. They both can define properties, methods and initializers.

Class examples

```
class PersonClass {
    let name: String
    var age: Int

    init(name: String, age: Int) {
        self.name = name
        self.age = age
    }
}

let instanceOfPersonClass = PersonClass(name: "Classy", age: 4)
let copyOfClass = instanceOfPersonClass

copyOfClass.name
copyOfClass.age

copyOfClass.age += 1

instanceOfPersonClass.age
copyOfClass.age
```

Difference between a class and struct

A Structure is a value type and Class is a reference type. When you copy a struct, you end up with two unique copies of the data. When you copy a class you end up with two references of the data. If class inheritance is not needed, structs are faster and more memory efficiency.

Biggest Difference

The biggest difference between class and struct is classes have hierarchical relationships. A chain of subclasses is called a hierarchy.

The major difference like class provides the flexibility of combining data and methods/functions and it provides the reusability called inheritance. Structs should be used for grouping data.

Class can create a subclass that can inherit parents properties and methods, where structure does not support the inheritance.