GENERAL ASSEMBLY

MOBILE DEVELOPMENT

LESSON 06
MORE CLASSES, STRUCTS, AND TYING INTERFACE BUILDER TO CODE

Tedi Konda
VP, Engineering and Technology, RepEquity

GETTING STARTED

READY?
SET?
GIT PULL!

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Object Oriented Principles Review
- Introduce structs
- Reference vs. Value Types Review
    - Classes and Structs
- In-Class Sample Project
- In-Class Group Assignment

GETTING STARTED

OBJECT ORIENTED PRINCIPLES REVIEW

WHAT ARE THE PRINCIPLES?

- 4 Principles
  - Encapsulation
  - Abstraction
  - Inheritance
  - Polymorphism

  - Examples in Playgrounds

ENCAPSULATION

- The implementation details (e.g., guts) of the function are hidden.

ABSTRACTION

- Abstraction in programming enables you, and enforces you to describe what you’re building as abstractly as possible.

INHERITANCE

- Inheritance is the idea that stating that something is a type of Animal gives it the properties of Animal.

POLYMORPHISM

- Polymorphism := “One Name, Many Forms”
  - A polymorphic concept in Swift is Function Overloading, which is having a function with the same name, but different parameters and return types.

STRUCTS

WHAT IS A STRUCT?

- Structures are blueprints of software constructs you want to build.
  - Typically, they are used when describing objects that have values.
    - Example: Rectangle

WHAT'S INSIDE OF A STRUCT?

- Properties
  - Constants and variables that describe the class
- Functions
  - Actions the function can perform with properties or other values.
  - Inside of Classes, functions are called methods.
- Memberwise Initialization
- Instances of Structs are not called Objects. They're simply called Instances.
  - To Playgrounds for examples on Structs

EXAMPLE OF A STRUCT

```
struct Rectangle {
    var x: Float
    var y: Float
    var width: Float
    var height: Float
}

/*
    Square is an instance of the Rectangle struct
    Square is created via memberwise initialization of the Rectangle
        struct, which means all the constants and variables are set
        during initialization.
*/
let square = Rectangle(x: 0.0, y: 0.0, width: 10.0, height: 10.0)
```

CLASSES VS STRUCTS

CLASSES VS. STRUCTS

- Classes
  - Mutable
  - Pass by Reference
- Structs
  - Immutable
  - Pass by Value
  - Memberwise Initialization

- To Playgrounds!

CODE-ALONG: COLOR WHEEL

GROUP PROJECT PROJECT: PLAYER SPRINT

IN-CLASS ASSIGNMENT

KEY OBJECTIVE(S)

Follow directions in lesson plan view controllers to create a sprint match between two players.

TIMING

40-50 min	1. Code with partner
5 min	2. Debrief

DELIVERABLE

Work in groups - ask questions if you need help!

HOMEWORK

HOMEWORK

- Read about:
  - Enumerated Types (e.g., enum, in the Enumerations chapter)
  - Switch Statements (in the Control Flow chapter)

  - Start Week 2 Homework
    - Found in Assessments folder
    - Due Sunday at Midnight

NEXT CLASS

NEXT CLASS

- Switch Statements (in the Control Flow chapter)
- Enumerated Types (e.g., enum, in the Enumerations chapter)
- View Controller Lifecycle
- Properties
- Gestures