SpaceAdventure
Lesson 4

Description

Add a SpaceAdventure class to the project, instantiate an object, and attempt to call an undefined method.

Welcome to our solar system!
There are 8 planets to explore.
You are currently on Earth, which has a circumference of 24859.82 miles.
What is your name?
Jane
Nice to meet you, Jane. My name is Eliza, I'm an old friend of Siri.
Let's go on an adventure!
Shall I randomly choose a planet for you to visit? (Y or N)
Huh?
Sorry, I didn't get that.
Y
Ok! Traveling to...

Learning Outcomes

• Recognize how classes can model and encapsulate concepts, and practice adding a Swift class definition to an existing Xcode project.

• Recognize compilation error notices the Xcode editor displays.

• Practice instantiating an object and calling a method.

Vocabulary

| modeling	object-oriented	class
Project Navigator	implementation	class definition
instantiate	object	method call

Materials

• **SpaceAdventure Lesson 4** Xcode project

Opening

How do you model real concepts in a Swift program?

Agenda

• Discuss the existing code in **main.swift**, and how the program is intended to be about a "space adventure," yet there is no "space adventure" in our code.

• Discuss the need to model what happens during the space adventure, such as greeting the travelers, asking them what planets they want to travel to, and then traveling to the planets.

• Discuss how Swift is an object-oriented language, and how one uses classes to define objects a program can use to model the concept of a space adventure.

• Add a new Swift file (⌘N) called **SpaceAdventure.swift**. Be sure that the SpaceAdventure group is selected, and that the SpaceAdventure target is checked.
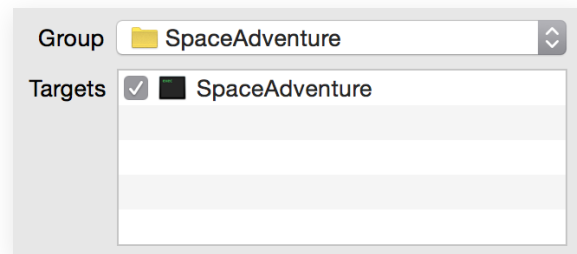
• Using the Project Navigator (⌘1), observe that Xcode displays the new file within the SpaceAdventure group.

• Explain the convention of using an individual file to contain a single class definition, and how the file name (**SpaceAdventure.swift**) alludes to the name of the class it contains.

• Discuss how, at a high level of thinking, the code in **main.swift** has just two jobs: to create a `SpaceAdventure` object, and to start the adventure.

• Above the existing code within **main.swift**, instantiate a `SpaceAdventure` object.

```
import Foundation

let adventure = SpaceAdventure()

let numberOfPlanets = 8
...
```

• Observe the errors in the Xcode editor.

• Discuss how we have yet to write the `SpaceAdventure` class definition, and therefore cannot create a `SpaceAdventure` object.

• Using the Project Navigator (⌘1), select **SpaceAdventure.swift** and implement a basic class definition.

```
class SpaceAdventure {

}
```

• Explain the class definition syntax, and the UpperCamelCase convention for class names.

• Return to **main.swift**, and observe that the error notice disappears.

• Discuss object instantiation syntax.

• Discuss how one might call a method upon a `SpaceAdventure` object, telling it to `start`.

• Add a method call using the `SpaceAdventure` object.

```
adventure.start()
```

• Observe the error notice in the Xcode editor.

Closing

Why won't our space adventure `start`?

Modifications and Extensions

• Explore the difference between Swift classes and structures, and replace the `SpaceAdventure` class definition with a structure.

• Explore the relationship between object instantiation syntax and initializers, and describe what initializers do in your own words.

Resources

The Swift Programming Language: About Swift https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/

The Swift Programming Language: A Swift Tour https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/GuidedTour.html

The Swift Programming Language: The Basics https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html

Project Navigator Help: Adding a New File https://developer.apple.com/library/ios/recipes/xcode_help-structure_navigator/articles/Adding_a_New_File.html

The Swift Programming Language: Classes and Structures https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ClassesAndStructures.html

The Swift Programming Language: Initialization https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Initialization.html

The Swift Programming Language: Methods https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Methods.html