General Assembly
Mobile Development

Lesson 3: Intro to Swift

Learning Objectives

- Define Swift and its value to the iOS ecosystem
- Define and demonstrate playgrounds
- Define Swift's fundamental data types
- Use variables and constants, and understand the difference between the two
- Apply optionals and understand when to use them
- Utilize control flow to create a simple program flow in playgrounds

Schedule

Time	Topic	Activity	Notes	Assessment
25 min	Questions & Review	Group review/quiz	It is important to engage the class in this exercise to re-iterate the high level functionality of IB, storyboards, nibs, view, and navigation controllers.	Students will answer all quiz questions as a group. They will explain a random answer to the class.
15 min	Agenda and introduction to Swift	Lecture & demonstration	Explain that we will work on integrating swift to our interfaces next class and that this class is focused on getting us familiar with the Swift programming language fundementals.	Students will be able to add a playground to their projects and print a hello world line with comments.

50 min	Types, variables and Constants	Demonstration (35 minutes) and guided practice with partner (15 minutes)	Playgrounds will be used to demo this section, as well as all others for the remainder of the class. **Need files**: InClassActivity.playground	Students will know the difference between variables and constants, name the basic data types, manipulate strings, know when to use optionals, and add type annotations.
60 min	Control flow	Demonstration (40 minutes) and guided practice with partner (20 minutes)	**Need files:** InClassActivity.playground	● Execute code based on certain conditions ● iterate through blocks of code ● the difference between a while loop and a for loop ● know how and when to use control transfer statements
15 min	Review and Q&A			

Questions & Review
Time: 25 Minutes

Topics	Interface Builder, Nibs, Storyboards, segues, navigation controller
Description	Break class up into groups answering all of the provided questions. Have each group explain answer to each of the quiz questions.
Activity Type	Group activity (class quiz):
Assessment	Students will answer all quiz questions as a group. They will explain a random answer to the class.

Instructional Design Notes

- Group activity rules:
    - One group for each question
    - Every group answers all questions (10-15 minutes to answer questions as a group)
    - After 5 minutes is up, randomly select each group to answer a specific question
- Questions:
    - What are the benefits of using nibs over storyboards and what are the downsides of storyboards?
    - What are the benefits of using storyboards over nibs and what are the downsides of nibs?
    - Give real world examples of nibs and storyboards (when one would use storyboards and nibs).
    - Why are segues important and how do you use them?
    - Define navigation controllers and give a sample use case.
    - Bonus question (either assign to specific group or ask to class at the end): What are outlets?

Link

Storyboards and nibs are great ways to layout app designs, however we still need to code the functionality and behavior of the app.

Use this as a transition point to explain why outlets are important and how they tie to code. This is where Swift, Apple's new programming language, comes in. Describe playgrounds and their significance in helping get instant feedback on code blocks.

Agenda and introduction to Swift

Time: 15 Minutes

Topics	Swift language, playgrounds
Description	Introduce class to swift. Give lecture on Swift history, benefits, and roadmap. Explain at high level why we need Swift for our applications. Demonstrate playgrounds.
Activity Type	Lecture and demonstration
Assessment	Students will be able to add a playground to their projects and print a hello world line with comments.

Instructional Design Notes

- Introduce Swift
    - Introduce Swift as a language for OS X and iOS.
    - Mention that Swift is an object oriented language. Briefly define object oriented.
    - Define typed languages and define Swift as a safely typed language.
    - Introduce Swift as a compiled language. Define the difference between scripted and compiled languages.
    - Swift is easier to get started with than Objective C. Mention that it looks like modern scripting languages, making it more intuitive to work with.
- Get students involved in helping identify types of programming languages. It is very likely that a good portion of students in the class will have experience with other programming languages and ecosystems.
- Compare Swift to other programming languages and list benefits.
- Explain the difference between the left and right panes in playgrounds.
- Introduce the println() method to build the first “Hello world” result.
- Introduce comments so that students can start annotating their code. Emphasize the importance of comments in code and mention that comments in code will count towards their evaluations.

Section activity (5 min)
Focused listing: instructor asks students to list out languages they may be familiar with and ask the students to define the languages as typed vs untyped and scripted vs compiled.

Types, variables, and constants

Time: 50 Minutes

Topics	Types, variables, constants
Description	Demonstrate basic data types, variables, constants, optionals, and type annotations. After the demonstration students will pair up and complete the provided playground file.
Activity Type	Demonstration and guided practice (with partner)

Assessment	Students will know the difference between variables and constants, name the basic data types, manipulate strings, know when to use optionals, and add type annotations.

Instructional Design Notes (35 min)

- Note that what we're about to talk about is new syntax. Swift syntax defines the rules for what constitutes valid Swift
    - All of our projects are comprised of syntactically valid Swift
    - The first half of the course we're going to be talking a lot about syntax
- Use slides as needed for each new concept, however the majority of the learning for this section will be done through playgrounds.
- **New Syntax:** Define and demonstrate the following data types on playgrounds: Booleans, Integers, Strings, Floats/Doubles.
    - Start with explicitly defining type annotations
- Introduce variables and constants and assign values from the data types mentioned above.
- It's important to explicitly state the difference between constants and variables. Make sure they are also familiar with mutable vs immutable concepts.
- Demonstrate how to add type annotations and explain how inferred type works.
- **Engage**: define variables and constants with certain values and ask the class what Swift infers the type to be.
- Demonstrate string interpolation and concatenation.
- Introduce optionals and demonstrate when to use them, and how to check for nil. Demonstrate how to unwrap optionals.
- While demonstrating optionals, give an example that results in a build error. Explain build errors and recognize that optionals help us identify errors on compile time. Build errors occur when we do not have valid syntax.

Section activity (15 min)

- Students to pair up for this exercise. Students will work together to complete the provided playground file.
- It's very important to check for understanding on optionals. Instructors and TA's to walk around during exercise and make sure that students are fully understanding optionals and the difference between variables and constants.

Control Flow

Time: 60 Minutes

Topics	Conditional statements, loops (for loop, while loop), control transfer statements (break, continue)
Description	Demonstrate control flow through provided code demo. At the end of the code-along students will pair up and complete the playground file.
Activity Type	Demonstration and guided practice (with partner)
Assessment	Students will complete exercise that demonstrates knowledge in: ● execute code based on certain conditions ● iterate through blocks of code ● the difference between a while loop and a for loop ● know how and when to use control transfer statements

Instructional Design Notes: guided practice (40 min)

- Introduce each new control-flow technique as **New Syntax**
- In the loops section highlight the different variations of both the for and while loops. Mention that the for-in loop will be very handy when we work with collections next week.
- Introduce increment/decrement operators, unary operators, remainder operator, and comparison operators.

Section activity (20 min)

Students will pair up for this exercise and work together to complete the playground file. It is critical to get a pulse on the class as they are working on this activity, as control flow will be essential next lesson. Instructors and TAs should walk around throughout the activity to gauge how students are doing. It's important to specifically check on loops throughout this exercise.
Activity bonus 1: Look up and use ternary operators where applicable in activity.
Activity bonus 2: Iterate through sample dictionary (mention that collections will be covered next week).

Review and Q&A

Time: 15 Minutes

Instructional Design Notes

- What is
- List the different Swift types.

- Highlight the difference between variables and constants. When to use each?
- What is a typed language? Is Swift typed?
- What is the difference between a compiled and scripted language? Which one is Swift?
- What are optionals? When would you use them?
- What is inferred type? What are type annotations?
- What is the difference between a for loop and a while loop?
- What's the difference between a for loop and a for-in loop?
- **Check for understanding (15 min)**
    - Instructors to randomly call students for each of these questions. Go into detail as needed for each question.

Link

Highlight that Swift will be the integral part of application logic and flow from this point on. Next class we will cover additional Swift functionality for better code efficiency and learn how to hook Swift logic into Interface Builder.