General Assembly
Mobile Development

Lesson 5: Making Interface Builder work with code

Learning Objectives

- Create hooks from interface builder to Swift code
- Create and implement custom classes
- Point out ability to access Xcode documentation for any external classes

Schedule

Time	**Topic**	**Activity**	**Notes**	**Assessment**
10 min	Questions & Review	Group discussion	Cover any questions from the last lesson.	
10 min	Class agenda and link to last session	Lecture	Explain that so far, our code and storyboards have not yet been connected	
75 min	Hook IB to Swift	Demonstration (45 minutes) and guided individual practice (30 minutes)	**Need files:** InClassActivity project	- Hook interface builder into Swift code - Update interface builder from swift code
75 min	Intro to classes	Demonstration (45 minutes) and guided practice with partner (30 minutes)	This is a very high level introduction to classes. Mention that we will go in much more detail next lesson.	- Identify and create classes
10 min	Review and Q&A			

Questions & Review

Time: 10 Minutes

Topics	Functions, parameters and return values
Description	Have students answer these questions in discussion format. They should use white board/live playground environment to make necessary points.
Activity Type	Class discussion
Assessment	Assess students’ knowledge of material through an oral quiz. Have students come up to explain and give examples of each of the quiz questions.

Instructional Design Notes
Questions:

- What’s a function? When might we use them?
- What is a parameter?
- What is a return value?
- What does the ‘return’ statement do in a function?
  How do we call a function without parameters? With parameters?
- Can functions have multiple return values? Multiple return statements?

Link (5 min)
Review last class’s material in a class discussion. Quiz the class on for and functions, parameters and return.

Hook interface builder to Swift

Time: 75 Minutes

Topics	View controllers, IB hooks (IB Outlets and IB Actions), update UI element properties
Description	It’s important to link to what we have learned so far and put into context that we have the basics of code and of interface builder down, so this is an important lesson to understand how to tie the two together.
Activity Type	Demonstration and guided individual practice
Assessment	Students will be able to hook interface builder into Swift code and update

	interface builder from swift code.

Instructional Design Notes

- Define view controllers and explain how to tie a view/scene to a view controller. Demonstrate how to hook up each view to an individual view controller.
- Demonstrate how to hook up UI elements to swift via IBOutlets and add functionality to buttons via IBActions.
- When hooking UI elements to the view controller show how to access documentation about specific classes with option+click and how to bring up the actual code for that class with command+click.
- Demonstrate how to access and change an element’s properties after an action.
- Introduce the toInt() method by grabbing user input in a text field and running a mathematical operation on it.

Section activity

students work alone on this exercise.

Instructions: create a new project with two scenes.

- Each scene will be tied to a different view controller.
- The first scene will display “Hello name” in a label when user hits the button. The name value will come from a text field element.
- The second scene will display the sum of two text fields in a label when user hits button.

Intro to classes

Time: 75 Minutes

Topics	Classes, instantiation, instance methods, properties
Description	Introduce classes as a blueprint for an instance. Mention that we will cover classes in much more detail next lesson.
Activity Type	Lecture/demonstration and guided practice (with partner)
Assessment	Students will be able to identify and create classes.

Instructional Design Notes (30 min)

- Identify classes as blueprint for code. Compare to how they can make Introduce class variables and demonstrate that their values help make your programs even more flexible than individual functions.
- Show class instantiation and give examples on how to call instance methods and get/set properties.

- Be sure to mention that methods are functions associated with a particular type (and class instances are of a class type).

Section activity (45 min)

- Students to pair up for this exercise.
- Students will work together to create a Math class that has the ability to add two numbers, multiply two numbers, and subtract two numbers.
- Students are to add a class variable of pi to be referenced from an instance or a blank variable

Bonus: Hook this class into IB functionality. Add two text fields, and a button for each of the operations. Each of the buttons will update the label.

Review and Q&A

Time: 15 Minutes

Instructional Design Notes

- What are view controllers and how do they impact your apps?
- How do you hook from IB to Swift?
- What is a class and what is an instance? What’s an instance method?
- **Check for Understanding (10 min)**
    - Check for understanding via “turn and talk”
    - Students to answer all questions as a team
    - Randomly call on a team to explain each question

Link

Mention that functions and classes are two of the most important concepts in Swift and many other programming languages. They make your code much more efficient and easier to work with. We now have the ability to add a lot of functionality to our designs.