GENERAL ASSEMBLY

MOBILE DEVELOPMENT

SWIFT CONTROL FLOW

William Martin
Head of Product, Floored

LEARNING OBJECTIVES

- Utilize “Control Flow” to make simple programs.
- Write conditional statements for logical decision-making.
- Write loops for basic automation and counting.
- Apply “Optionals” and explain when and how to use them.

AGENDA

- Logistics (10 min)
- Review exercises from last class (10 min)
- Questions about Assessment 2 (5 min)
- Show your work! Assessment 1 (20 min)
- Activity 1 - Conditionals, loops, and Optionals (15 min)
- Break (5 min)
- Activity 2 - Questions addressed (15 min)
- Slides - 30 min
- Activity 3 - Code-Along (30 min)
- Follow-Up Material + For Next Class (10 min)
- Exit tickets (10 min)

REVIEW VALUES, TYPES, VARIABLES, CONSTANTS

- Convert 100ºC to Fahrenheit
- Prove that all the other planets can fit between the Earth's surface to the Moon's surface.
- Make a generic monthly budget. Compute savings and the rent you can afford for different salaries.

ASSESSMENT 2 QUESTIONS

ASSESSMENT 1 - SHOW YOUR WORK!

I NEED 3 VOLUNTEERS...

ACTIVITY 1

KEY OBJECTIVE(S)

Review the pre-work and describe conditionals, loops, and Optionals to each other.

TIMING

14 min	1. Ask questions of each other about conditionals, loops, and Optionals.
1 min	2. Post questions to Slack.

DELIVERABLE

Write some questions down and post to Slack. The TAs and instructor will collate and address them after a short break.

PROMPTING QUESTIONS

- What is an if-statement? How do they work? Write an example that uses an “else” clause.
- What do loops do? Why would you use one?
- What are the different kinds of loops? What makes them different?
- When would you use an Optional? In what kinds of situations would they be useful?
- How would you sum all even numbers from 0 to 1000?
- Write a single question you’d like to ask the instructors (or the class).

CONTROL FLOW

CONTROL FLOW

CONTROL.PLAYGROUND

CONTROL FLOW

- Programs are executed one line at a time, but it's not useful to execute all lines of code all of the time.
- Conditional statements leverage Boolean expressions to begin to define the logic of our apps. We can execute some code under certain conditions, and other code under other conditions.

CONTROL FLOW

- We can start to reason like this:
  - e.g. “If the temperature is less than or equal to 32 degrees, show a freezing icon, otherwise, show water drop icon.”
- Also, we can start to leverage a computer’s automation abilities by using loops.
  - e.g. “Keep executing this code as long as the temperature is less than 32.”

CONTROL FLOW - CONDITIONALS

Conditional statements, or “if-else” statements, look like this:

```
if temp <= 32 {
    // This “block” is executed if the condition is true.
    // Show a freezing icon.
} else {
    // And this “block” if false.
    // Show a water drop icon.
}
```

CONTROL FLOW - CONDITIONALS

Conditional statements can contain multiple blocks or clauses, using “else if”:

```
if temp <= 32 {
    // Show a freezing icon.
} else if temp >= 212 {
    // Show a boiling water icon.
} else {
    // Show a water drop icon.
}
```

CONTROL FLOW - WHILE LOOPS

The simplest kind of loop, while loops execute a block of code repeatedly as long a given condition is true.

```
var sum = 0
while sum < 50 {
    sum += 10
}
println(sum)
```

CONTROL FLOW - FOR LOOPS

Strangely named, “for-loops” use conditionals to continue executing code given a conditional and a variable that is used for counting.

```
for (var temp = 0; temp <= 32; temp++) {
    // Do something here.
}
```

CONTROL FLOW - FOR LOOPS

```
for (var temp=0; temp<=32; temp++) {
    // Do something here.
}
```

1. The loop declares and initializes a variable (temp),
2. checks the conditional, and if it's true,
3. executes the block of code within the braces, then
4. calls the incrementing expression (temp++)
5. checks the conditional again, etc.

CONTROL FLOW - CONTROL TRANSFER - BREAK

```
let toCheck = 289
for (var i=2; i<toCheck; i++) {
    println(i)
    if toCheck % i == 0 {
        println(“composite!”)
        break
    }
}
```

The “break” statement aborts from the for loop.

Advanced students: make this more efficient. Write as a while loop.

CONTROL FLOW - CONTROL TRANSFER - CONTINUE

```
let toCheck = 289
for (var i=2; i<toCheck; i++) {
    if i % 2 == 0 { continue }
    if toCheck % i == 0 {
        println(“composite!”)
        break
    }
}
```

The “continue” statement skips everything after it in the block, but continues executing the loop.

OPEN OPTIONALS.PLAYGROUND

OPTIONALS AND NIL

- `nil`
  - A value that represents no value.
- Optional - a type that represents `nil` or a value of another specified type
- Syntax:

```
var [symbol] : [type]?
```

- Example

```
var name : String?   // initialized as nil
var name : String? = “Toshi”
```

OPTIONALS AND NIL

- Why use Optionals?
  - Sometimes we need a variable before we get a chance to give it a real value.
  - e.g. A user profile that treats the user's middle name as optional.
  - e.g. Imagine a web request that takes some time. We need a place to put the response to that query, but we won't know what the response is until the request is done.

OPTIONALS - UNWRAPPING

- Optionals have two somewhat incompatible states:
  - nil, representing no value
  - has a value of a particular type
- In order to get to an Optional's value (if it's not nil), we have to "unwrap" by adding an ! right after the variable:
  - `var name : String?`
  - `name = "Toshi"`
  - `println("My pup's name is \(name!).")`

OPTIONALS - UNWRAPPING

- However, there's a problem with unwrapping. You can't unwrap an Optional if it's nil. In that case, the syntax `name!` would cause an error.
- How do we deal with this? This syntax helps us distinguish between the nil and value-holding cases, and also unwraps the value if it's available (i.e. not nil):

```
if let _name = name {
    println("The pup's name is \(_name).")
} else {
    println("I don't know the pup's name…")
}
```

ACTIVITY 1

KEY OBJECTIVE(S)

Exercise conditionals, loops, and Optionals as a class.

TIMING

25 min	1. Code along with the instructor.
5 min	2. Go over solutions together.

DELIVERABLE

Deliver some variations on the exercises due next class.

EXERCISE – WHOSE LINE IS IT ANYWAY?

- Scenario:
  - Build a weather app that suggests what to wear based on the temperature outside.
- Conditionals: What to wear based on a temperature range.
- Variation: Use Optionals to describe known/unknown temperature.

EXERCISE – COUNTING WITH LOOPS

- Add numbers from 1 to 10.
- Loop over numbers from 1 to 100, count the multiples of 3 or 5, but not numbers divisible by 3 and 5.
- Sum all the numbers in a multiplication table.

- Variation: Sum all the prime numbers from 1 to 1000.

FOLLOW-UP MATERIAL

- Apple's Documentation
  - Control Flow: https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ControlFlow.html

FOR NEXT CLASS

- *We Heart Swift*
  - Chapter 7 on Functions
  - Stop before "Constant and Variable Parameters."
- Apple's Documentation
  - Functions: https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Functions.html
  - Stop before "Functions with Multiple Return Values."

REVIEW