Announcements

• Final Project descriptions are due on Friday

• Final Project Presentations start on Monday
  – I’ll send an email out this weekend on the presentation order

Today’s Topics

• Swift
  – Overview
  – Syntax
  – Examples

• Xcode 6
  – Playgrounds

• Swift code in Objective-C

• Objective-C code in Swift
Swift

- New programming language developed by Apple

- Announced at WWDC 2014

- Interoperates with Objective-C
  - Both are considered first class citizens

- Still a work in progress

- Latest version is Swift 2.1
  - Released on Sept 23rd 2015

Hello World in Swift

```swift
print("Hello World")
```

- No semicolons

- No main method needed
Variables and Constants

- **Swift uses** `var` and `let` **to describe variables and constants**

- **Variables and constants have a type**
  - `let` `languageName`: `String` = “Swift”
  - `var` `version`: `Double` = 1.0
  - `let` `isEverChanging`: `Bool` = true

- **Swift supports type inference**
  - `let` `languageName` = “Swift” //inferred as `String`
  - `var` `version` = 1.0 //inferred as `Double`
  - `let` `isEverChanging` = true //inferred as `Bool`

Strings

- **Swift makes working with strings easy**
  
  let `firstName` = “John”
  let `lastName` = “Smith”
  let `fullName` = `firstName` + “ “ + `lastName`

- **Enumerating through them is familiar**
  
  for `character` in `firstName`.characters{
      print(`character`)
  }

  J
  o
  h
  n
String Interpolation

let a = 2, b = 3

// “2 times 3 is 6”
let mathResult = “\(a) times \(b) is \(a * b)”

Collections - Arrays and Dictionaries

var names = [“Bob”, “Alice”, “Mike”, “Jen”]
– Inferred as a typed collection of Strings

• I could also be more explicit:
  var names: [String] = [“Bob”, “Alice”, “Mike”, “Jen”]

var numberOfLegs = [“ant”: 6, “snake”: 0, “cow”: 4]
– Inferred as a typed dictionary of Strings and Ints

• Or I could be more explicit:
Loops

while !done {
  keepDoingSomething()
}

for num in 1...5 {
  print("(num) times 4 is (num * 4)"
}

for var i= 1; i<= 12; i++ {
  doSomething(i)
}

//Prints from 1 up to and including 5

Conditionals

if legCount == 0 {
  print("Does not walk")
} else if legCount == 1 {
  print("Hopping around")
} else {
  print("I can walk")
}

switch legCount {
  case 0:
    print("Does not walk")
  case 1, 3, 5, 7:
    print("Limps around")
  default:
    print("I can walk")
}
**Functions**

```swift
func sayHi() {
    print("Hi")
}
sayHi()

func sayHi(name: String) {
    print("Hi \(name)!")
}
sayHi("Bob")
sayHi(name: "CSE 436")
```

**Functions**

```swift
func sayHi(name: String = "CSE 436") -> String {
    return "Hi " + name
}
let name = sayHi()  // Name contains "Hi CSE 436"

func refreshWebSite() -> (Int, String) {
    // refresh
    return (200, "Success")
}
let (statusCode, message) = refreshWebSite()
```
Closures

- Self-contained blocks of functionality that can be passed around
  - Similar to blocks in Objective-C

```
let displayGreeting = {
  print(“Hello Class”)
}

let displayGreeting: () -> () = {
  print(“Hello Class”)
}

displayGreeting()
```

Optionals

- Optionals handle the absence of a value
  - There is a value and it equals x or there isn’t a value

```
var numberOfLegs = [“ant”: 6, “snake”: 0, “cow” :4]
let possibleNumLegs = numberOfLegs[“goat”] ???
let possibleNumLegs: Int? = numberOfLegs[“goat”] //Value or nil

If possibleNumLegs != nil {
  let legCount = possibleNumLegs! //Use ! to unwrap the optional
  print(“Goat has \(legCount) legs”)  //Use \( to format string
}

- Shorthand for above, if let

If let legCount = possibleNumLegs {
  print(“Goat has \(legCount) legs”)  //Use \( to format string
```
class Person {
    //properties
    //methods
    //initializers
}
• No .h files
• No Universal base class //Can use if needed

class Person {
    var age = 21  //defines the properties
    var description: String {  //defines a computed property
        get {
            return “You \(\text{age}\) years old”
        }
    }
}

let somePerson = Person()  //no alloc needed
print(“Hello, you are \(\text{somePerson.age}\) years old”)
More Information about Swift Language

- Swift Programming Guide
  - Available on iTunes
- WWDC 2014 and 2015 Videos
  - developer.apple.com

Examples in Playground
Integrating Objective-C with Swift

Integrating Swift with Objective-C
iOS App Development Opportunity

Alex Agatstein – alex.agat@gmail.com