Programming and Development Concepts

Cindy Royal, Ph.D
Making A Mobile App
School of Journalism and Mass Communication
Texas State University

Intro to Programming

- Introduce more advanced functionality; interactivity, data presentation, device specific
- Many programming languages, each with different purposes
- Machine/assembly languages are coded directly to the processor
- Interpreted languages are more easily written and read by humans. Interpreter converts it to machine language – ex. JavaScript
- Compiled languages send code to compiler that creates an executable native to the hardware – ex. C
- Object-oriented – program contains objects which are reused throughout – ex. Java
- Server-side languages which execute on the server while the user is interacting – Java, PHP, Ruby, Python
- Client-side languages that execute on the user's computer – JavaScript
- Mobile development languages – Swift, Java

Intro to Programming

- While languages are different, many concepts are shared
- May use slightly different syntax

Variables

Allows you to store a value so it can be used throughout the program

```
var name = “Cindy”

var num = 2

var score:Int = 0
```

Some programs require that you use the prefix when assigning a variable, some don’t. Some require you to define the type (Int, text, etc).

Data Types

- Text values are known as strings and are usually encased in quotation marks
- Numeric values can be integers or with decimals – either float or double
- Each programming language has unique ways to deal with data types
- Boolean is also a data type, set as true or false

Concatenation

- Use concatenation to join strings, values and variables
- Some languages use the “+” sign for concatenation

```
first = “Cindy”;
last = “Royal”;
message = “Hello “ + first + “ “ + last
```

- With Swift, you can use slash notation to concatenate a variable to text:

```
myScore.text = "Score: \(score)"
```

Properties and Methods

- Most languages have properties or methods that help you add functionality
- For example, the length property finds out how many characters are in a string.

```
password.length
```

- If you assign a word to the password variable, for example, “myterriblepassword”, the length would be 18.
- Length includes special characters and spaces
- Swift uses the ”count” property

```
password.characters.count
```

If Statements

if statements introduce logic by providing a decision point or condition and what to do depending on the response to the condition

```
if(counter == 0) {
                timerLabel.text  =  "Time's  Up!"
                timer.invalidate()
          }
```

Notice the use of two equal signs for the condition. This indicates comparison. Use one equal sign when you are assigning a value to a variable, as in the timerLabel text.

Loops

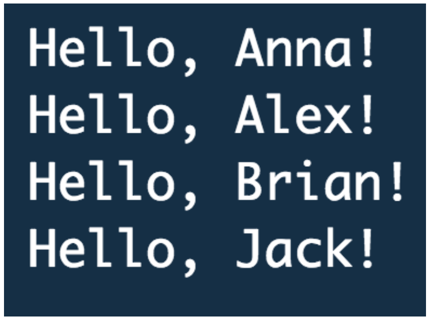
Loops are a different kind of logical situation. They continue as long as a condition is met. Once the condition is no longer true, the loop ends. It allows the program to repeat instructions over and over again until the condition is no longer true.

```
let names = ["Anna", "Alex", "Brian", "Jack"]
for name in names {
  print("Hello, \(name)!")
}
```

Loops

Loops are a different kind of logical situation. They continue as long as a condition is met. Once the condition is no longer true, the loop ends. It allows the program to repeat instructions over and over again until the condition is no longer true.

```
let names = ["Anna", "Alex", "Brian", "Jack"]
for name in names {
  print("Hello, \(name)!")
}
```



```
Hello, Anna!
Hello, Alex!
Hello, Brian!
Hello, Jack!
```

Functions

- Functions allow you to store lines of code that can be accessed and used again and again throughout a program.
- Here's an example of a Swift function we'll be using in the next lesson

```
func didScore(points:Int) {
        score = score + points
        myScore.text = "Score: \(score)"
}
```

- Functions can take arguments that get passed into the function so it uses different assumptions each time it is run.
- When a function is called, the argument is passed in the place of the parameter

```
didScore(points:5)
```

Arrays or Lists

- An array or list is a way to store multiple items

```
var shoppingList: [String] = ["Eggs", "Milk"]
```

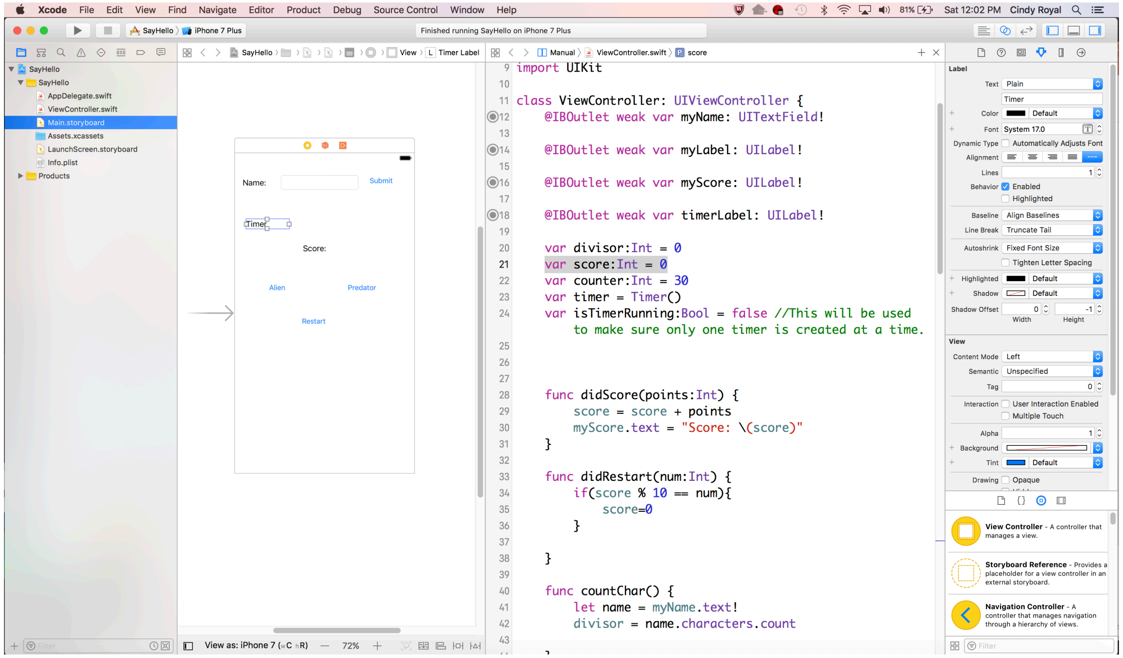
- You can can access elements of an array by using the index number in bracket notation. Arrays start with index 0.

```
print(shoppingList[0])
```

- This command would return “Eggs”, the first item in the list

Where do you code?

- Text editor
- Interactive Development Environment
- In a development console
- Xcode – combo of visual tools and coding


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
81% Sat 12:02 PM Cindy Royal
SayHello 〉 iPhone 7 Plus
Finished running SayHello on iPhone 7 Plus
SayHello
SayHello
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
SayHello 〉 View 〉 Timer Label
Name:
Submit
Timer
Score:
Alien
Predator
Restart
View as: iPhone 7 (wC hR)
72%
Manual 〉 ViewController.swift 〉 score
Filter

```
import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var myName: UITextField!

    @IBOutlet weak var myLabel: UILabel!

    @IBOutlet weak var myScore: UILabel!

    @IBOutlet weak var timerLabel: UILabel!

    var divisor:Int = 0
    var score:Int = 0
    var counter:Int = 30
    var timer = Timer()
    var isTimerRunning:Bool = false //This will be used
        to make sure only one timer is created at a time.



    func didScore(points:Int) {
        score = score + points
        myScore.text = "Score: \(score)"
    }

    func didRestart(num:Int) {
        if(score % 10 == num){
            score=0
        }

    }

    func countChar() {
        let name = myName.text!
        divisor = name.characters.count
```

Label

Text: Plain
Timer
Color: Default
Font: System 17.0
Dynamic Type: □ Automatically Adjusts Font
Lines: 1
Behavior: ☑ Enabled
□ Highlighted
Baseline: Align Baselines
Line Break: Truncate Tail
Autoshrink: Fixed Font Size
□ Tighten Letter Spacing
Highlighted: Default
Shadow: Default
Shadow Offset: Width 0, Height -1

View

Content Mode: Left
Semantic: Unspecified
Tag: 0
Interaction: □ User Interaction Enabled
□ Multiple Touch
Alpha: 1
Tint: Default
Drawing: □ Opaque

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Final Thoughts on Programming

- Programming is problem solving
- Programs usually don’t work the first time
- Your job is to figure out and troubleshoot the problem and fix it to get your desired functionality
- There are many resources online. Google your problem, find examples.
- Don’t let the code beat you! Have fun.

More information on programming

- You have access to Lynda.com as a TXST student. Login on the TXST Lynda page.
- The Computer Science Principles: Programming course can provide more info about general programming concepts. Check it out. Sections 1-3 will give you more details about what I covered.