CS342/MED253 Building for Digital Health
Lecture 2A: Basics of iOS Development (using Swift with xcode) and Git

Welcome!

https://cs342.stanford.edu
cardinalkit.slack.com

Don’t forget to record lecture
Overview for today

- Announcements

- Swift Basics

- What is SwiftUI?

- Coffee Time (breakout in groups of 3 for socializing)

- Live-code demo.

https://cs342.stanford.edu

cs342-aut1920.slack.com

Don’t forget to record lecture
Please read the following abstracts 🤝

PROJECT A

Development of App to Facilitate Post-Kidney Transplant Outpatient Care
Led by Marc L. Melcher, MD, Ph.D., Department of Surgery (melcherm@stanford.edu)

The goal of this project is to develop an interactive mobile app that facilitates the complex outpatient care of kidney transplant patients to increase medication compliance and to improve management of comorbidities, both of which are essential to the ultimate success of kidney transplantation. To achieve this, the app would provide patient education and collect important post-operative data. The data will be used to create bi-weekly reports that will guide the post-operative care of patients for the first 30 days after transplantation.

PROJECT B

Digital Health Cardiovascular Disease Management System
Led by Paul J. Wang, MD and Meg Babakhanian, PhD (pjwang@stanford.edu, mbabakha@stanford.edu)

The goal of the project is to leverage digital health technology to create a platform for management of a range of cardiovascular diseases, from prevention to disease management. The project will create a web-based platform as the provider-facing user interface and patient-facing Apps for patient engagement using their iPhones to allow for cardiovascular disease management. ECGs from Apple Watch will be used to document the electrocardiogram.

---

Project A
Noah Jacobson
Kabir Jolly
Rachel Naidich
Amrita Kaur

Project B
Henry Mell sop
Colton Swingle
Collin Schlager
Assignment #1: Hello World 👋
Set up Xcode, create a basic SwiftUI app, and submit via GitHub.

Prerequisites for Developing on iOS

You will need access to a mac for development. If you do not have a mac, the macs in residential clusters and at Lathrop (for checkout) come with Xcode pre-installed. You may also try a cloud solution such as macincloud.

Xcode
iOS app development.
Please download version 12.3 or higher via developer downloads or via the app store.

Things to note:
- Install command line tools if prompted.
- Login with your AppleID
- We will focus on creating and leveraging SwiftUI apps.

Due Jan 26th
For this assignment, your goal is to set up Xcode, create a basic SwiftUI app, and submit via GitHub.

You will be graded for completion of the following:

1. Install Xcode and Git on your machine.
2. Complete Chapter 1 of the official SwiftUI tutorial.
3. If you are new to Git, complete the following tutorial.
4. Create a SwiftUI iOS app with a basic information card about yourself (85%) 
5. Submit your code via a GitHub pull request (15%)
Due Jan 26th

For this assignment, your goal is to set up Xcode, create a basic SwiftUI app, and submit via GitHub.

You will be graded for completion of the following:

1. Install Xcode and Git on your machine.
2. Complete Chapter 1 of the official SwiftUI tutorial.
3. If you are new to Git, complete the following tutorial.
4. Create a SwiftUI iOS app with a basic information card about yourself (85%)
5. Submit your code via a GitHub pull request (15%)
let’s go through some basics!

refer to this guide
Constants (`let`) and variables (`var`)
Type Annotations

```swift
let maximumNumberOfLoginAttempts: Int = 10

var welcomeMessage: String = "Hello"
```

Int, Float, Double, Bool, String
Comments & Print statements

1 // This is a comment.
2
3 /* This is also a comment
4 but is written over multiple lines. */
5
6 let languageName = "Swift"
7
8 print(languageName)
9 // Prints "Swift"
10
11 print("The current value of languageName is \(languageName)")
12 // Prints "The current value of languageName is Swift"
Booleans and if-statements

```swift
let orangesAreOrange = true
let turnipsAreDelicious = false

if turnipsAreDelicious {
    print("Mmm, tasty turnips!")
} else {
    print("Eww, turnips are horrible.")
}
// Prints "Eww, turnips are horrible."
```
Optionals

1 var convertedNumber: Int? = 404
2 //convertedNumber contains an actual Int value of 404
3 convertedNumber = nil
4 //convertedNumber now contains no value

var possibleNumber = "123"
2 var convertedNumber = Int(possibleNumber)
3 //convertedNumber is inferred to be of type "Int?", or "optional Int"
Optionals

```swift
1 if convertedNumber != nil {
2     print("convertedNumber has an integer value of \(convertedNumber!).")
3 }
4 // Prints "convertedNumber has an integer value of 123."

1 if let convertedNumber = Int(possibleNumber) {
2     print("The string "\(possibleNumber)" has an integer value of \(convertedNumber)"")
3 } else {
4     print("The string "\(possibleNumber)" could not be converted to an integer")
5 }
6 // Prints "The string "123" has an integer value of 123"
```

Arrays

```swift
var shoppingList = ["Eggs", "Milk"]
print("The shopping list contains \(shoppingList.count) items.")
shoppingList.append("Flour")
shoppingList[0] // Eggs
```
Dictionaries

```swift
1 var airports: [String: String] = ["YYZ": "Toronto Pearson", "DUB": "Dublin"]
2 var airports = ["YYZ": "Toronto Pearson", "DUB": "Dublin"]
3
4 print("The airports dictionary contains \(airports.count) items.")
5 // Prints "The airports dictionary contains 2 items."
6
7 airports["LHR"] = "London"
8 // the airports dictionary now contains 3 items
9
10 airports["LHR"] = "London Heathrow"
11 // the value for "LHR" has been changed to "London Heathrow"
```
For-In loops

```swift
let names = ["Anna", "Alex", "Brian", "Jack"]
for name in names {
    print("Hello, \$(name)!")
}
// Hello, Anna!
// Hello, Alex!
// Hello, Brian!
// Hello, Jack!
```
For-In loops

```swift
let numberOfLegs = [
    "spider": 8,
    "ant": 6,
    "cat": 4
]

for (animalName, legCount) in numberOfLegs {
    print("\(animalName)s have \(legCount) legs")
}

// cats have 4 legs
// ants have 6 legs
// spiders have 8 legs
```

https://cs342.stanford.edu
# Structures and Classes

```swift
struct Resolution {
    var width = 0
    var height = 0
}

class VideoMode {
    var resolution = Resolution()
    var interlaced = false
    var frameRate = 0.0
    var name: String?
}
```
Structures and Classes

```swift
let someResolution = Resolution()
let someVideoMode = VideoMode()

print("The width of someResolution is \(someResolution.width)")
// Prints "The width of someResolution is 0"

print("The width of someVideoMode is \(someVideoMode.resolution.width)")
// Prints "The width of someVideoMode is 0"

someVideoMode.resolution.width = 1280
```

https://cs342.stanford.edu

© 2016 Stanford Byers Center for Biodesign
Functions

```swift
1 func greet(person: String) {
2     print("Hello, \\$(person)!")
3 }
4
5 func greet(person: String) -> String {
6     let greeting = "Hello, " + person + "!"
7     return greeting
8 }
9
10 func greet(person: String, alreadyGreeted: Bool) -> String {
11     ...
12 }
```

Int, Float, Double, Bool, String
Enums

```swift
enum CompassPoint {
    case north
    case south
    case east
    case west
}

var directionToHead = CompassPoint.south

switch directionToHead {
    case .north:
        print("Lots of planets have a north")
    case .south:
        print("Watch out for penguins")
    case .east:
        print("Where the sun rises")
    case .west:
        print("Where the skies are blue")
}

// Prints "Watch out for penguins"
```

https://cs342.stanford.edu
SwiftUI

Building a modern user interface for your digital health app
UIKit

The predecessor of SwiftUI
Imperative Avocado Toast

1. Prepare ingredients: avocado, bread, almond butter, sea salt, red pepper flakes.
2. Gather equipment: toaster, plate, butter knife.
3. Lightly toast a slice of bread.
4. Place toast on plate.
5. Spread thin layer of almond butter over toast.
6. Cut avocado in half on its longer axis.
7. Remove avocado core.
SwiftUI

The modern, *declarative* way to design user interfaces for iOS
Declarative Avocado Toast

“I’d like some avocado toast on charred sourdough with almond butter, sea salt, and red pepper flakes.”

“Oh, and cut it diagonally.”

“Please and thank you!”
Writing in SwiftUI feels natural (source: AppCoda)
SwiftUI provides an immediate connection (source: AppCoda)
“Creators need an immediate connection to what they create. And what I mean by that is when you’re making something, if you make a change, or you make a decision, you need to see the effect of that immediately.”

- Bret Victor
# Declarative
small_nums = [x for x in range(20) if x < 5]

# Imperative
small_nums = []
for i in range(20):
    if i < 5:
        small_nums.append(i)

That Python example IS NOT declarative.  – Juanjo Conti  Nov 23 '09 at 18:06

Prolog is a declarative programming language.  – Juanjo Conti  Nov 23 '09 at 18:10

@Juanjo: It IS declarative. – missingfaktor  Apr 10 '10 at 5:27

How is the first statement in here any more declarative than the second?  – zenna  May 9 '12 at 21:46

Agreeing with Juanjo and zenna - a loop construct does not magically transform into a declarative program when refactored into a shorter notation.  – Felix Frank  Jun 14 '14 at 21:29

Disagreeing with @FelixFrank, and leaning towards @missingfaktor's "bold" statement. The traditional, "fully" declarative way to do this is `filter(lambda x: x < 5, range(20))`, is just another refactoring into a shorter notation. This is not in any meaningful way different from the list comprehension expression (which has clear "map" and "filter" sections), which was created (see pep 202) with the explicit intention to create a more concise notation. And this list comprehension would be more clear/idiomatic in this case.  – yoniLavi  Mar 20 '15 at 11:38

As I see it, if something is declarative or not will depend on the way you code it, and not on how it's actually executed/compiled. If the code you are writing describe what you want and not how to achieve it, then is declarative. So, even if the call is a simple construct, it may be still declarative. At the end everything is compiled into processor instructions, that is always imperative.  – JP Ilienes  Jul 9 '15 at 9:21

LINQ is not declarative. It may look like it, but it is still executing the expression left to right exactly as written.  – Jonathan Allen  Oct 15 '15 at 17:54

@JonathanAllen that's certainly true when dealing with an IEnumeratable, but in the case of IQueryable, a chain of statements may be in fact "executed" at the same time and in a different order, as is the case when Linq ends up generating SQL. Since exactly the same statements would be written in either case, I would consider most of Linq a good example of declarative style. Our knowledge of the implementation details shouldn't change the "declarativeness" - if the code is expressing intent but not explicit implementation, then in the vast majority of cases it would be considered declarative.  – Rob Hill  Jul 27 '17 at 11:42

At the risk of committing the genetic fallacy, I would say list comprehensions are eminently declarative constructs, they come from Haskell, which is a purely functional programming language, and that is unambiguously in the declarative paradigm. The fact the there is a for-loop and `.append` underneath the hood is not relevant. You can always go deeper and find, fundamentally, an imperative program due to the nature of common computer architectures.  – juanpa.arrivillaga  May 1 '20 at 7:41
How can you use SwiftUI to supercharge your iOS workflow?

**Goal:** be able to write basic SwiftUI and fluently read SwiftUI codebases

**Sources:** goshdarnswiftui.com, developer.apple.com/documentation/swiftui/
Views and Controls

# Text

A view that displays one or more lines of read-only text.

Text("Hello World")

Styling

Text("Hello World")
  .bold()
  .italic()
  .underline()
  .lineLimit(2)
Views and Controls

# Button

A control that performs an action when triggered.

```javascript
Button(
    action: {
        // did tap
    },
    label: { Text("Click Me") }
)
```

If your `Button`'s label is only `Text`, you can initialize with this simpler signature.

```javascript
Button("Click Me") {
    // did tap
}
```
Views and Controls

Show map with a specified region

```swift
import MapKit

@State var region = MKCoordinateRegion(center: .init(latitude: 37.334722, longitude: -122.085747), span: MKCoordinateSpan(latitudeDelta: 0.1, longitudeDelta: 0.1))

Map(coordinateRegion: $region)
```
# HStack

A view that arranges its children in a horizontal line.

To create static scrollable List

```swift
HStack (alignment: .center, spacing: 20){
    Text("Hello")
    Divider()
    Text("World")
}
```
Layout and Presentation

# VStack

A view that arranges its children in a vertical line.

To create static scrollable List

```swift
VStack (alignment: .center, spacing: 20){
    Text("Hello")
    Divider()
    Text("World")
}
```
Links to Bookmark

https://learnxinyminutes.com/docs/swift/

https://goshdarnswiftui.com

Let’s review some basic git commands...

# SETUP & INIT

Configuring user information, initializing and cloning repositories

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>git init</code></td>
<td>Initialize an existing directory as a Git repository</td>
</tr>
<tr>
<td><code>git clone [url]</code></td>
<td>Retrieve an entire repository from a hosted location via URL</td>
</tr>
</tbody>
</table>
## STAGE & SNAPSHOT
Working with snapshots and the Git staging area

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>git status</code></td>
<td>show modified files in working directory, staged for your next commit</td>
</tr>
<tr>
<td><code>git add [file]</code></td>
<td>add a file as it looks now to your next commit (stage)</td>
</tr>
<tr>
<td><code>git reset [file]</code></td>
<td>unstage a file while retaining the changes in working directory</td>
</tr>
<tr>
<td><code>git diff</code></td>
<td>diff of what is changed but not staged</td>
</tr>
<tr>
<td><code>git diff --staged</code></td>
<td>diff of what is staged but not yet committed</td>
</tr>
<tr>
<td><code>git commit -m &quot;[descriptive message]&quot;</code></td>
<td>commit your staged content as a new commit snapshot</td>
</tr>
<tr>
<td><strong>BRANCH &amp; MERGE</strong></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Isolating work in branches, changing context, and integrating changes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>command</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>git branch</strong></td>
<td>list your branches. An * will appear next to the currently active branch.</td>
</tr>
<tr>
<td><strong>git branch [branch-name]</strong></td>
<td>create a new branch at the current commit.</td>
</tr>
<tr>
<td><strong>git checkout</strong></td>
<td>switch to another branch and check it out into your working directory.</td>
</tr>
<tr>
<td><strong>git merge [branch]</strong></td>
<td>merge the specified branch’s history into the current one.</td>
</tr>
<tr>
<td><strong>git log</strong></td>
<td>show all commits in the current branch’s history.</td>
</tr>
</tbody>
</table>
How would you build this?

COVID-19 Screening Tool
This tool can help you understand what to do next about COVID-19.

Let’s all look out for each other by wearing a mask, avoiding crowds, and by knowing our status.

Start Screening

State Information
Choose a state to see guidance from the health department.

More Information
About COVID-19
An up-to-date rundown of the virus and its symptoms.

What You Can Do
Coffee Time

- 5 - 10 min
- Opportunity to mingle / take a break
- Ice breaker question for the day:

  How would you build this?

https://cs342.stanford.edu

cs342-aut1920.slack.com
COVID-19 Screening Tool

This tool can help you understand what to do next about COVID-19. Let's all look out for each other by wearing a mask, avoiding crowds, and by knowing our status.

Important Message

Local Testing Center

SwiftUI Basics

Build a simple SwiftUI interface!
ResearchKit is an open source framework introduced by Apple that allows researchers and developers to create powerful apps for medical research.

Assignments
#2: Assignment #2: ResearchKit + Firebase
Please read the following abstracts 🎓

PROJECT A

Development of App to Facilitate Post-Kidney Transplant Outpatient Care
Led by Marc L. Melcher, MD, Ph.D., Department of Surgery (melcherm@stanford.edu)

The goal of this project is to develop an interactive mobile app that facilitates the complex outpatient care of kidney transplant patients to increase medication compliance and to improve management of comorbidities, both of which are essential to the ultimate success of kidney transplantation. To achieve this, the app would provide patient education and collect important post-operative data. The data will be used to create bi-weekly reports that will guide the post-operative care of patients for the first 30 days after transplantation.

PROJECT B

Digital Health Cardiovascular Disease Management System
Led by Paul J. Wang, MD and Meg Babakhanian, PhD (pjiang@stanford.edu, mbabak@stanford.edu)

The goal of the project is to leverage digital health technology to create a platform for management of a range of cardiovascular diseases, from prevention to disease management. The project will create a web-based platform as the provider-facing user interface and patient-facing Apps for patient engagement using their iPhones to allow for cardiovascular disease management. ECGs from Apple Watch will be used to document the electrocardiogram.

Project A
Noah Jacobson
Kabir Jolly
Rachel Naidich
Amrita Kaur

Project B
Henry Mellsop
Colton Swingle
Collin Schlager
Attendance Check 🙋🏼

- Use the following link to mark your attendance for today:

https://cs342.stanford.edu
Thank you!