In this lab, we’ll cover 5 things,

1. Git
2. CocoaPod
3. Segue between ViewControllers
4. Understand the lifecycle of a ViewController
5. How to use delegate

Now let's start with

1. Git

The first thing you should do is to start with registering an account of Github: https://github.com

After registration, start with creating a new repository, for public repository, it’s free, but anyone can see your code.

Now after you create your new repository, here's an example https link to your git

https://github.com/xuaninbox/CS407.git

So you have two ways to link your git with your project, with command line or with GUI interface.
here’s how you can do it with GUI in Xcode

switch to the **remote** tab and **add a remote**
again in the **Source control**, click **commit**

![Image](https://via.placeholder.com/150)

*UR NOTE*

- Push to remote:  
  - WHATEVER/master (Create...)

A **Message** is required and when select **Push to remote**, it will also push it to the remote git.

**Now go back to ur git and see if ur code is there**

### 2. CocoaPod

CocoaPod is a easy way for u to manage the libraries of your project, now start with the tutorial to install cocoapod on [https://cocoapods.org](https://cocoapods.org)

**after you finish the INSTALL section,**

use command line to go to ur project folder

and do **pod init**

**go on with the GET STARTED section BUT**

instead of using the example Podfile in the **GET STARTED** section, use the following

```
target 'P2' do
  pod 'AFBlurSegue', '~> 1.2.1'
end
```

**after pod install** go to you project, from now on you should use **XXX.xcworkspace**
Awesome! You just installed your first 3rd party library: AFBlurSegue

https://github.com/AlvaroFranco/AFBlurSegue

3. Segue

In this section you will create the two basic segue, Show and Present modally

Now use 3 View controller
The two button (show & present modally) will let you either show the show or present modally to the new controllers.

Also by using

```swift
override func prepareForSegue(segue: UIStoryboardSegue!, sender: AnyObject!) {
    if (segue.identifier == "XX") {
        // pass data to next view
    }
}
```

You should make the label on the new controllers to print ‘SHOW’ or ‘MODAL’
Now try run it in the simulator and see if it works!

4. lifecycle

This should be a easy one! lets try to understand how the view controller works, try to do the following

```swift
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        print("viewDidLoad")
    }

    override func viewWillAppear(animated: Bool) {
        print("viewWillAppear")
    }

    override func viewDidLoad(animated: Bool) {
        print("viewDidAppear")
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}

Now try run it in the simulator and tell us what is the sequence of the print and why so.

5. Delegate

Delegate is one of the most important aspect of the iOS programming, lets start with a simple delegate, UITextFieldDelegate, this delegate is used to handle text UITextField.
Let's say you have a UITextField on your ViewController and you want to be notified every time user starts to type something in it.

First drag the UITextField onto any of the ViewControllers you have created, and set the component's delegate as its ViewController.
Add the delegate in your swift code and the function that handle the callback

```swift
import UIKit

class ViewController: UIViewController, UITextFieldDelegate {

    override func viewDidLoad() {
        super.viewDidLoad()
        print("viewDidLoad")
    }

    override func viewWillAppear(animated: Bool) {
        print("viewWillAppear")
    }

    override func viewDidAppear(animated: Bool) {
        print("viewDidAppear")
    }

    func textFieldDidBeginEditing(textField: UITextField){
        print("textFieldDidBeginEditing")
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

As you can see we have added `UITextFieldDelegate` and a function `textFieldDidBeginEditing`

Now try run it in the simulator and see what happened when you start typing into the textfield.