Lab 1 - Hello World

Introduction

The objective of this Lab is to confirm that you have properly installed Xcode, have access to Github.com and can create a simple iOS application. You should have completed the study of iOS 14 Getting Started, modules 1 and 2 before attempting this assignment. This is your first programming assignment in this class, so there are a lot of detailed steps, but no code. Going forward, the assignment writeup will not contain as many details as by then you will know what we mean by instructions like "commit" and "push to Github".

Strings in all caps such as YOUR_GITHUB_USERID in the instructions below means you need to substitute the appropriate string.

Install Xcode On Your Late-Model Mac

This should already be completed during your study of iOS 14 Getting Started modules 1 and 2.

You should say yes to installing the additional required components (you might be prompted at Xcode startup)

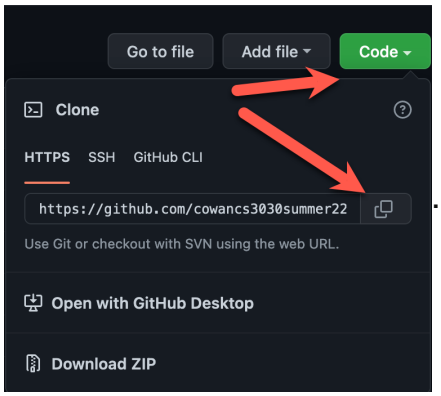
Add Your Github Account to Xcode

Add your Github credentials to Xcode so that Xcode can access your repos on Github.com. This step needs to be performed only once this semester.

- Create an account on www.github.com if you don't already have one. I suggest that you choose something professional-sounding and not something that might give the wrong impression to future employers.
- On github.com, right-click on your profile icon and click Settings. Click on Developer settings and then Personal access tokens. Give your token a name (such as Xcode or something memorable), either a long expiration date or no expiration at alll, and select all rights and click Generate token. Copy the token to your clipboard. I recommend saving this token in a password database for future use.
- In Xcode, go to Preferences→Accounts and add a Github account with your username and token and sign in. Close the Accounts window.

Generate a Private Github Repository

Github Classroom will generate a private repository (repo) for you to use to submit this assignment.

- Go to the Canvas Module for this assignment, click on the Clone This Repo link and follow the instructions to accept this assignment and receive a private repo. You may be required to log into github.com.
- Github Classroom will create and assign to you a source code repository (repo) with a link of the form: HTTPS://github.com/cowancs3260<SEMESTER>/helloworld-YOUR_GITHUB_USERID.git.
- Once your repo has been created (refresh the page until you see your repo), click on it. You will be taken to your repo on github.com
- On github.com, click on the green Code button and you will see the name of your repo and a clipboard icon to the far right of it 

. Click on this clipboard icon  to copy your repo name to your clipboard. Every repo has two names on Github, one beginning with HTTPS and one beginning with git@github.com. Either will work. HTTPS requires less setup. If you are curious, ask me in Ted Cowan Live why I never use the HTTPS link.
- Remember that the name of your Hello World repo is on your clipboard, which you will use in a later step.

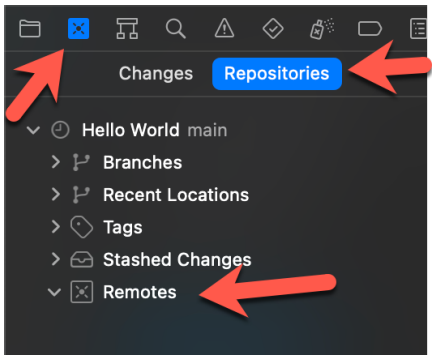
Create Your initial Hello World project

- Launch Xcode and create a new project. Select iOS and App as the template and click Next.
- Set the product name to exactly “Hello World” and note the capitalization and the single space between Hello and World.
- Make sure that "Include Unit Tests" and "Include UI Tests" are checked (different from what you saw in the videos) and the language is set to Swift.
- Select SwiftUI as your User Interface. We will not be using Storyboard in this class.
- Click Next.
- When saving the project make sure Source Control is checked. Click Create.
- For this assignment, to make the instructions easier to follow, save your Hello World project onto your Desktop. If you can remember where you stored your project, feel free to save it anywhere.

Adding Your Remote Github Repo To Your Project

Adding a remote repo to your project tells Xcode where to send your project code when you request a "Push". Doing a `git push` begins the grading process.

- Click on the Source Code Navigator icon, expand Hello World master and right click on Remotes:



- Select “Add existing remote”. Leave Remote Name as "origin".
- If your repo name is still on your clipboard, paste it in the Location field. Otherwise, enter in the full path to your Github Classroom-supplied repo. If you aren’t using a clipboard manager, I highly recommend one. Ask me about it in the Instructor’s Blog.
- Click Add.

Share Your Scheme

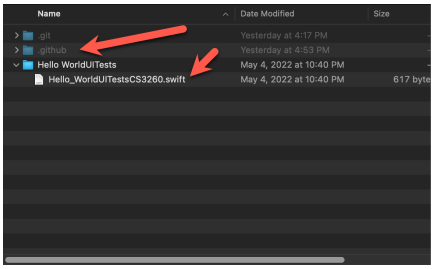
Share your scheme so your program can be compiled by our Jenkins testing server:

- Go to Products→Schemes→Manage Schemes and locate the Hello World scheme (it should be the only one). Ensure there is a checkmark in the Shared box on the far right and close the window
- Select the iPhone 11 in the upper left of the Xcode screen (the tests are geared for the iPhone 11).

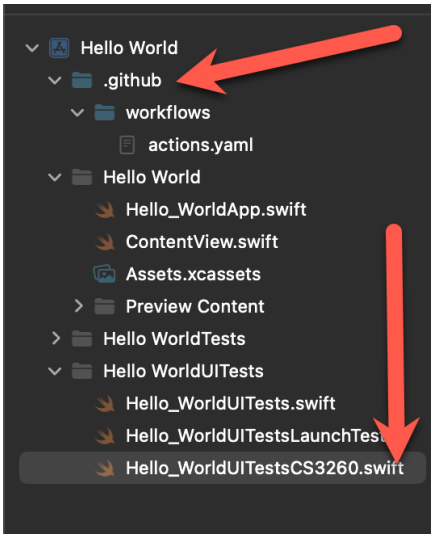
Add Support Files and Commit Them to Github

To enable testing and grading of your lab, two files must be added to your Hello World project.

- Your Xcode window should still be open to your Hello World project. Click the Project Navigator icon (the folder in the upper left).
- Open Terminal and issue command cd Desktop
- Enter command git clone https://github.com/tcowan/cs3260helloworldtemplate.git
- Open Finder and navigate to the cs3260helloworldtemplate folder on the Desktop. If needed, expand the Hello WorldUITests folder.



- Drag the `.github` folder (NOT THE .git folder) from the Finder window to the very top of the Xcode Navigator pane, just under the topmost “Hello World”. Make certain the "Copy items if needed" box is CHECKED.
- Drag Hello_WorldUITestsCS3260.swift from the template to the Xcode Navigator pane, just under the folder Hello WorldUITests. Make certain the "Copy items if need" box is CHECKED. After dragging the .github folder and the UITests files from Finder to Xcode, you should be able to see this in the Xcode Navigator:



- Double-check that you placed Hello_WorldUITestsCS3260.swift in the Hello WorldUITests folder and nowhere else and the .github folder is in the right place.
- Select menu option Source Control → Commit… to save your changes to Git. Add a commit message like "added support files". Select the checkbox at the bottom left that says "Push to remote:" and click the Commit button.
- CMD-R to build and run. Ensure the Simulator pops up with your Hello, World app.

Commit Your Label Change to Github

- Click menu item Source Control→Commit and look near the bottom for an unchecked box next to your scheme. Check the box.
- In the bottom window, enter “Initial commit” (or whatever you would like) as your commit message.
- Select the checkbox at the bottom left that says "Push to remote:" and click the Commit button.
- Congratulations! By pushing your code to the remote repo, you have turned it in for grading. Nothing needs to be done in Canvas to indicate you have turned in your assignment.
- Drum roll please: you have completed your first iPhone app assignment!

How Grading Works

Your grade is automatically posted to Canvas each time you push your code to Github. The only way to receive a grade for a coding assignment in this class is to push it to your provided repo, prior to the deadline.

About ten minutes (up to an hour) after pushing to Github, you should receive an email to your Weber State email account from "Github Actions", containing a report of the compile and your test results. The tests that are graded are contained in the source file Hello_WorldUITestsCS3260.swift. That file contains an assertion that your app contains a Text field containing the text "Hello, world!" and if that test passes, you will receive full credit for the assignment.

We will talk more about how this testing magic works (especially when it doesn’t 😊) as we move forward this semester.

If your code contains errors and you push that code to Github, no points will be posted to Canvas and you will receive an email with a file called consolelog.txt. It will state that the build of your source has failed. This file is usually included in every email to the student just in case something bad happens and we need to troubleshoot.

Although we are using a semi-automated grading tool, your instructor retains the right at any time to inspect your code, run additional tools and tests and to adjust your grade accordingly.

Do not modify the template files in any way, other than to place them as indicated in this assignment.

Here Is How You Earn Points For This Assignment:

FEATURES	POINTS
Required Features	
App compiles with no errors	0
App displays a Text field containing exactly the words "Hello, world!"	100
Grand Total	100