Mobile 325 Prework

Most assignments listed below are to complete the actual lesson or project listed as if you were a scholar. Read the corresponding column for context or additional guidance on what to take into consideration before starting each lesson/project. That column will also contain focus questions for you. By the time you complete each lesson, you should be able to answer those! Lastly, each week's work includes links to external resources that we've vetted and found to be helpful. You can, of course, find additional resources to supplement your learning if that is helpful to you, as well.

Graphic Organizers: In the intro video, we asked that you keep your notes for your prework all in one dedicated notebook. In addition to your notes, we recommend using this graphic organizer (print out or "Make a Copy" to your Google Drive) *or* creating a similar one somewhere in your notebook! This is intended to keep you focused on those key ideas and understandings of the content (and should probably help when you start thinking about visual aides in your classroom!) At different points throughout the prework, the deliverables will ask for you to share a picture or screenshot of your graphic organizer.

Assignment Due Dates all by 9 pm MST (3+ hours per Technical assignment):

- Technical 1 due March 29
- Technical 2 due April 5
- Technical 3 due April 12
- Technical 4 due April 19
- Technical 5 due April 26
- Technical 6 due May 3
- Technical 7 due May 10

Instructors who start camp after June 15th should also complete (1-2 hours per challenge):

- Complete Challenge 1 during the week of May 10-17
- Complete Challenge 2 during the week of May 17-24
- Complete Challenge 3 during the week of May 24-30

Find all lessons and projects here: https://turingschool.github.io/kwk-curriculum-site/login/

Technical 1	
Assignment	**Details**
Super Learner	Read this article and think about what steps you need to take to fully engage in this learning process.
Install Xcode	Install Xcode from the app store on your Mac. It is recommended you do this before heading to bed or not needing your computer for several hours - it's a big application! You may be required to update your MacOS before you are able to download. *The curriculum was built with the expectation that you are running on Xcode version 11.3.1 which is compatible with any Mac running on macOS 10.14.4 or later.*

	Get familiar with Xcode Playgrounds by reading this (we recommend you stop at the Running Your Code in Xcode section) Note: You do need to have a relatively current version of Xcode to do most of this prework. - If you have an older version, but are able to complete this week's pre-work, please DM Amy Holt on Slack ASAP with the screenshot of your MacOS version and Xcode version. - If you are not even able to download Xcode for whatever reason, please DM Amy Holt on Slack ASAP. We will find a way to get you the machine you need but will need time so must hear from you immediately! For the Week 1-2 pre-work, you can use one of the following online tools:https://repl.it/, http://online.swiftplayground.run/, https://www.weheartswift.com/swift-sandbox/
What is Mobile Development? What is Swift? *Highly Recommended*	We're all coming into this with different levels of experience. You may not know much about coding at all; you may have a lot of experience, but aren't sure what type of coding we'll be doing this summer. This 8-minute video explains what Mobile Development is at a high level and starts to get into what that will mean for our work this summer.
Intro to Swift Lesson	If you don't have an iPad or iPhone to complete the Warm Up with - it's ok! You can look at any similar to-do list app and still get the takeaways from the activity. Focus Questions: - What is Swift and why do we need it? - What are two data types? - What is interpolation?
Conditionals Lesson	Focus Questions: - Provide an example of where/when a real web application that probably uses a conditional - List 3 operators and explain what each of them does. Which is the newest or more interesting to you? - What is the difference between = and ==?
Functions Lesson	Focus Questions: - In your own words, how would you explain what a Swift function is? - What is the difference between declaring and calling a function? - What is an argument?
Optional Additional Resources	● Video on Variables ● Video on Data Types ● Video on IF statements ● Intro Video on Functions ● Video on Functions (return values and arguments)
Deliverables	**Share the following in your prework channel:** **➔ Video Reflection** **◆ Introduce yourself** **◆ 'What are your key takeaways from this week's prework?** **◆ Did you get stuck at any point?**

	◆ **What did you do when you got stuck or felt challenged?** ◆ **What ways could you improve your study/work habits in the next week?** ➔ **Written responses to Conditionals Focus Questions**

Technical 2	
Assignment	**Details**
Arrays Lesson	Focus Questions: - What is the purpose of arrays? - List all the ways you can currently interact with the data in an array
Dictionaries Lesson	Focus Questions: - How would you explain the difference between an array and a dictionary? When would you use one as opposed to the other? - Why are dictionaries so much more powerful than Strings or Ints?
Looping Lesson	Focus Questions: - What is the purpose of a for-in loop? - How many times will the code inside of a for-in loop run? - Where might an application that you use, implement something like a for-in loop? Explain.
Optional Additional Resources	- Dictionaries Chapter - Arrays Chapter - Video on Arrays - Blog Post (stop when you get to While loops)
Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **What did you do when you got stuck or felt challenged?** ◆ **Have you been able to leverage the additional resources?** ◆ **Is there a benefit in looking at other resources while you are learning this content?** ➔ **Written responses to Looping Focus Questions** ➔ **Your graphic organizer/notes where you are keeping track of Swift vocabulary, data types, keywords and programming concepts (screenshot or photo upload)**

Technical 3	
Assignment	**Details**
Classes + Objects Lesson	This class is not slated to be explicitly taught at camp. If you and your partner Instructor want to add it in, you absolutely can! The rationale: this lesson helps build some foundational understanding that will provide value to you as you move into more advanced work with Swift. In past years, scholars weren't excited about this content. With the short amount of time at camp, they weren't able to see this foundation as helpful. However, since instructors have a longer time to learn Swift, we want to help you build a stronger foundation. This will

	take you one step closer to some deep understanding that will help you problem solve and answer scholar questions when you are at camp. Focus Questions: - What is the difference between a class and an object? - What is the purpose of a class?
Object Interaction Lesson	Same as above - this class is not slated to be explicitly taught at camp. Focus Questions: - How would you explain to a 5 year old - what is a property and what is an action? (in relation to objects) - When we are starting out, we use physical objects to model because it helps build a mental model. Thinking of an application you use - where do you see a template being used over and over, but each instance has slightly different info? If you wrote a class for that template, what properties and actions might it have?
Array Challenges	Solve the "YELLING" Array challenge. If you need a refresher on working with arrays or want to see an example of someone solving one, watch this video first! *Optional: Here is another set of challenges* *https://codewithchris.com/learn-swift/#challenges*
Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **Why do you think we are having you do prework that isn't going to be taught at camp?** ◆ **How did you approach the "Yelling" Array challenge? What was your plan? Did it go as you expected?** ◆ **What would you do differently if we gave you that same challenge again?** ➔ **Your graphic organizer/notes where you are keeping track of Swift vocabulary, data types, keywords and programming concepts** ➔ **A code snippet of the code you wrote to solve the "YELLING" Array exercise.**

Technical 4	
Assignment	**Details**
Xcode Projects Lesson	Get organized: Up until now, most of the code you've written was for the sake of learning, and you've improved so much that you could easily re-create. You are going to start building projects that you may want to refer back to. Decide where you are going to save them on your computer and the naming conventions you will use to organize all the projects. Make sure you feel comfortable with the following: - Expanding and collapsing the Navigation Pane, Debug Area, and Utilities Pane. - Expanding and collapsing theDocument Outline when a StoryBoard is

	being viewed. - Running and stopping the simulator - Switching between viewing a .swift file and a StoryBoard - Switching the device and orientation that you are viewing on a StoryBoard
StoryBoards Lesson	Watch each video and complete the required activities to familiarize yourself with StoryBoards. If time permits, feel free to start another project or two and let your curiosity guide you to discover other parts of Xcode projects! Focus/Reflection Questions: - What benefit do constraints provide? - List at least 4 types of Objects from the Object Library and describe them.
Actions + Outlets Lesson	Focus/Reflection Questions: - What is the utility of an action? Outlet? - Think back to the “Reminders” App that we discussed in the Warm Up of the Intro to Swift lesson (or a similar type of app). Draw out a sketch of the app, and mark it up with places that you think actions and outlets are likely used.
Git & GitHub (under “Other”)	Walk through this tutorial and get familiar with using these features! Then, we recommend using this Git Workflow on a regular basis so it becomes second nature to you. You may need to download Git on your machine - you can do that here. FYI - there are many ways that developers interact with Git! We believe this to be the most straightforward method to use, limiting time scholars have to take away from learning Swift and building awesome projects.
Optional Additional Resources	As you look for resources to better navigate Xcode projects and build a StoryBoard, don’t be discouraged, but be aware: Apple iterates very quickly on this software, so any video or tutorial older than 6 months may show a screen with a slightly different UI. They will likely still be helpful, just go into that carefully!
Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **How did you feel as you started your project?** ◆ **Did you take time to create a plan before you jumped in to coding? Did the plan help you?** ◆ **What do you think will be important for your scholars to know, think, feel as they start this project?** ➔ **Written responses to/photos of Actions + Outlets Questions** ➔ **A code snippet of the code from your View Controller file that has at least one action and one outlet. Follow up with a short explanation of why you needed each respective action and outlet.**

Technical 5	
Assignment	**Details**

Segues	This lesson is designed to a) teach some technical content and b) give scholars some independent problem-solving and self-teaching skills! Work through the steps in the lesson as listed out so that YOU also get both benefits. *We've mentioned a couple times that some documentation will be out of date due to Apple updating software frequently, but that it can still be helpful. The blog post linked here is a good example of that - the UI screenshots may differ a bit from yours, but the gist of what is being taught is still relevant. We are pointing this out because you will have to google and research a lot on your own - before and during camp and want to support you in developing that sense of what is helpful and what is not.*
Tables	This lesson is more like a step-by-step tutorial. In past years, many scholars have wanted to build apps with a table-like view, so we wanted this resource to be available to them. By working through this, you'll be more prepared to support them and very likely gain some valuable debugging skills! *Building a table like this is not something you should expect yourself to memorize how to do.*
Emojional (Mini Project)	This is another opportunity to use all those pieces of knowledge you have, and put them together in (possibly) a slightly different way. You should be able to complete Iteration 2 on your own at this point. Attempt Iteration 3 but don't be discouraged if you need to reach out for help while working through it!
Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **What is your problem solving process?** ◆ **Were you able to use those skills in your Seques lesson?** ◆ **How did you develop your problem solving skills in your life?** ◆ **What support did you need from other people to develop those skills?** ➔ **Your graphic organizer/notes where you are keeping track of Swift vocabulary, data types, keywords and programming concepts**

Technical 6	
Assignment	**Details**
Core Data Code Along - coming soon!	Focus Questions: - Question 1 - Question 2 - Question 3
To-Do List (Mini Project)	Complete The To-Do List Mini Project. Follow the Git workflow laid out in the project so that you get that practice and can share your work with us!
Optional Additional Resources	Add in if we find any - concern about versions of Swift/Xcode causing problems like last year

Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **How did you engage yourself during the Code Along? Did you end up just "Sitting & Gitting?"** ◆ **What was effective about the Code Along approach?** ◆ **What more could you have used as a student to really dig into the content?** ➔ **Written responses to Core Data Focus Questions** ➔ **The link to your GitHub repository for the To-Do List Project**

Technical 7	
Assignment	**Details**
Accessing Camera + Displaying Photos	Like Tables, this is not something you should expect yourself to memorize how to do! A professional developer would definitely have to google and look up examples, etc. in order to implement this. We want you to have the experience of struggling through that so you are better positioned to support your students in that productive struggle. This is also more of a tutorial than a lesson as this utilizes libraries that must be used in very specific ways.
Start ViewFinder	Complete Iterations 1 and 2 of the ViewFinder Project. If time permits, start on Iteration 3, but it is not required at this time. Follow the Git workflow so you can share your progress!
Deliverables	**Share the following in your prework channel:** ➔ **Video Reflection** ◆ **Reflect back on your 8 weeks of prework experience…** ● **What did you learn about yourself?** ● **What habits and practices helped you be successful?** ● **How do you want to translate this to your camp?** ➔ **The link to you GitHub repository for the start of ViewFinder** ➔ **Your rough drafts for at least 2 anchor charts you want to use in your classroom at camp for: Swift vocabulary, data types, keywords, and/or programming concepts (upload photos/screenshots)**