(http://bitfountain.io/course/complete-ios8)

Lemonade Stand Assignment!!!

(mailto:john@bitfountain.io)

Objective:

It’s now time to dive into a major project that we’re going to call Lemonade Stand. One of the best ways to learn to program is to try to build the challenges on your own. This is going to test every concept that we have gone over up until this point. Consequently, this might seem intimidating and overwhelming at first, but just remember to work on each section at a time. However, I can guarantee when you complete this project, it is going to boost your coding confidence and get your mind accustom to thinking like a programmer!

Let me remind you this is meant to be a challenge. However, if you break up the problem into smaller chucks, it will make the goal much more attainable. This is why we STRONGLY recommend doing this. You can follow our recommended steps, or break it into your own.

Furthermore, we encourage you to challenge yourself even further and customize your application and attempt it own your own before watching the solution videos. Bear in mind, the solution video is not the only way to solve the assignment. If your application meets the requirements set below in a different approach, then you have solved the problem.

The solution we have provided is a clean and concise version that may provide to be a useful and helpful comparison as you work through the assignment.

Project Requirements:

This application is meant to be a simple game so you should keep this in mind if you are planning to implement your own custom design. The goal of the game will be to maintain a profit each day of the game. As soon as you run out of money to purchase new inventory, the game should end.

Here are some more details that will be important for the game. The player should start with $10, 1 Lemon, and 1 Ice Cube. Each additional Lemon will cost $2, and each additional Ice Cube $1.

First thing that you should do is set up the storyboard. We've attached a screenshot of the view, so that you can imitate it for your project. Be aware of the spacing and connection type of each object in your view. We don't want anything overlapping or looking cluttered.

Also, make sure you are hooking up your storyboard elements properly. For example, your declared IBActions, for when a button is pressed, should be of connection type, **Action**, with type **AnyObject**, with the Event type, **Touch Up Inside**, and finally, has the argument, **Sender**.

As for the storyboard setup, make sure you have the following 4 sections: What supplies you (the player) have, a section to purchase supplies (like more lemons, ice cubes, etc.), a third area where you can create the lemonade, and lastly, a section with your sell lemonade button.

Screenshot of finished Storyboard:

FinishedScreenShot (https://www.dropbox.com/s/wf29527mxpl3dvt/Finished%20Storyboard%20Screenshot.png?dl=1)

Recommended Steps and Hints:

1. Setting and connecting the storyboard elements, which includes:
    1. Labels
    2. Buttons
    3. IBOutlets
    4. IBActions
    5. Any other UI elements you want to add…
2. Then, make it so that we are able to purchase, and un-purchase, lemons for $2 and ice cubes at $1. When the user presses the corresponding "+" and "-" buttons.
3. Add mix/un-mix buttons for your lemonade that day that will be added into your lemonade for the day, they should only be mixed when the day officially starts. Very similar to step #2.
    1. Remember to subtract the added lemons and ice from your inventory amount when

adding them to your mix.

4. When we hit “Start Day” Button, it should create a lemonade ratio that will affect sales, (lemons over ice cubes), i.e. more acidic, equal parts, or less acidic lemonade.
   1. For example, if you add 1 lemon and 3 ice cubes the ratio will be .333.
5. Still under the start day button, create a random number of customers (between 1 and 10) that will visit your stand for the day.
6. Then, create a random taste preference(between 0 and 1) for each customer (hint, this should be a constant and you may want to use a type of loop to generate each preference).
   1. For example, you might generate 5 customers for the day with preferences as such: 0.5, 0.7, 0.3, 0.4, 0.1
   2. Then, you should compare your preferences to a range of values, as well as the current lemonade ratio to a separate range of values. Then see if they match. We are **not** comparing the ratios directly to each other! Ranges are shown in the steps below. ***So, what do we mean!? Here's a breakdown:***
      1. You should compare your randomly-generated customer's preferences to 3 different ranges from 0 to 1.
         1. The three ranges are:
            1. 0 to 0.4 – favors acidic lemonade
            2. 0.4 to 0.6 – favors equal parts lemonade
            3. 0.6 to 1 – favors diluted lemonade
      2. At the same time, you should be comparing your lemonade ratio to a different set of ranges.
         1. Greater than 1 (Acidic Lemonade)
         2. Equal to 1 (Equal Portioned Lemonade)
         3. Less than 1 (Diluted Lemonade)
      3. Here's where it will come together. I will spell out the first set of logic and hopefully you'll be able to do the rest.
         1. **For example**, If customerPreference is **Less than** 0.4 **AND** lemonadeRatio is

                  Greater than 1... Get paid a $1 and print out "Paid!"

                  2. This should read, If the customer likes acidic lemonade (between 0 – 0.4) and the lemonade is acidic (greater than 1) Get paid a dollar!
                  3. Follow this example for the next 2 comparison sets...
                  4. At the end, you'll want an **else** statement where they didn't want to buy your lemonade, so you will not be paid.
                  5. **You'll want to run through this logic for each customer you randomly generate.**

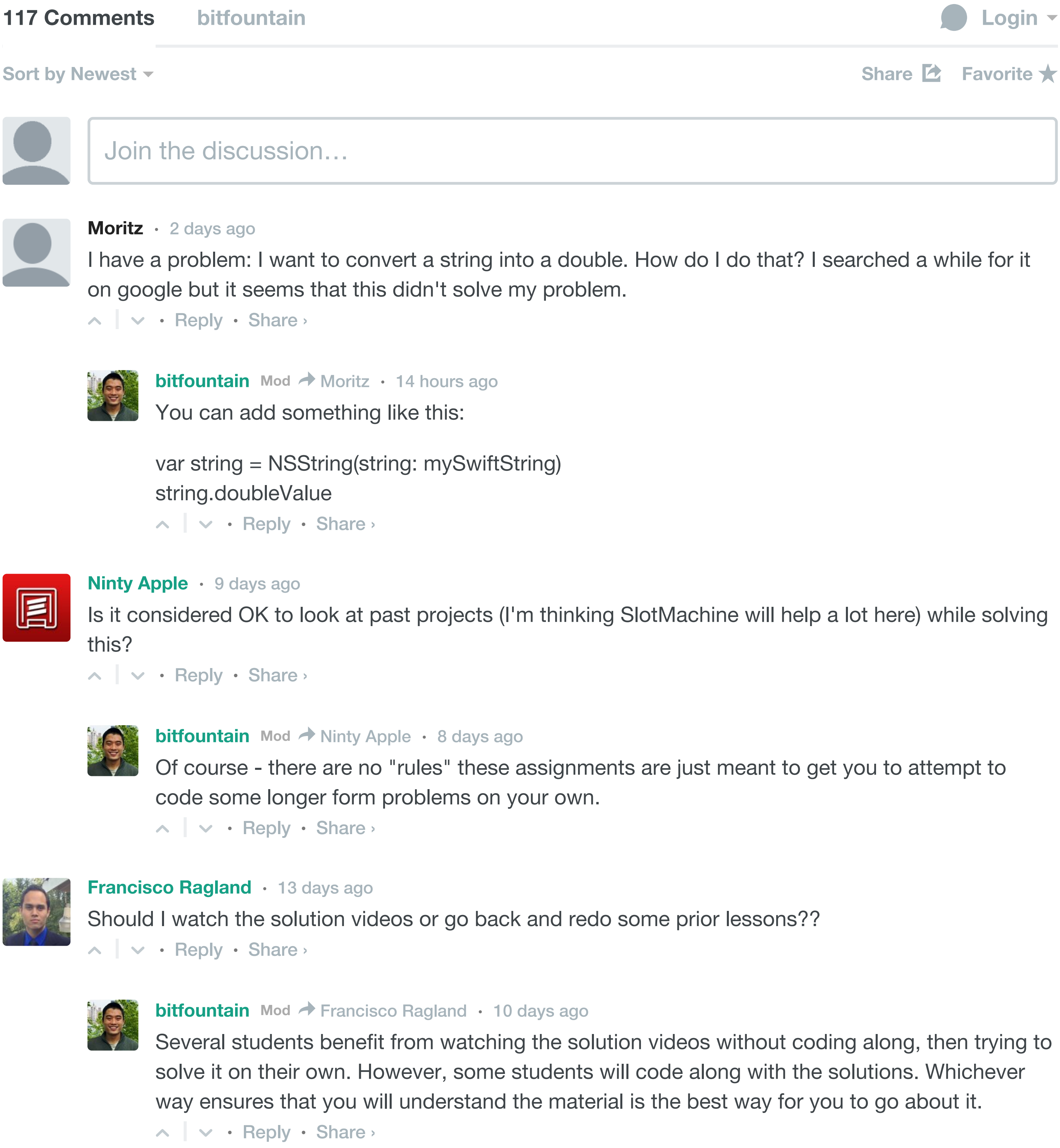
7. While you are comparing the preferences to these ranges, we want to see which customers are buying and which are not. So, write a **println** that will show
    1. each customer’s generated preference value, and
    2. whether or not they matched to the lemonade.
        1. If they matched, print “Paid!”
        2. If not, print “No match, No Revenue.”

Bonus Challenge:

1. Now, not everyone will every stop for lemonade. Especially if it’s raining. So, Display a random Weather status using image assets that will affect how many customers show up (Colder weather = less customers, Warmer weather = more customers).
    1. For example, cold weather means the random amountOfCustomers – 3
    2. Fair weather means no change,
    3. and then maybe warmer weather = amountOfCustomers + 4.
        1. Feel free to play around with these numbers.

Also, here are the image assets to use to display each weather type. Be sure to use a UIImageView.

Lemonade Weather Assets: Download Assets (https://www.dropbox.com/sh/489qphm2hiityle/AADcxu2l1bpbsoHBMuoLP7MLa?dl=1)



Moritz · 2 days ago

I have a problem: I want to convert a string into a double. How do I do that? I searched a while for it on google but it seems that this didn't solve my problem.

bitfountain Mod → Moritz · 14 hours ago

You can add something like this:

```
var string = NSString(string: mySwiftString)
string.doubleValue
```

Ninty Apple · 9 days ago

Is it considered OK to look at past projects (I'm thinking SlotMachine will help a lot here) while solving this?

bitfountain Mod → Ninty Apple · 8 days ago

Of course - there are no "rules" these assignments are just meant to get you to attempt to code some longer form problems on your own.

Francisco Ragland · 13 days ago

Should I watch the solution videos or go back and redo some prior lessons??

bitfountain Mod → Francisco Ragland · 10 days ago

Several students benefit from watching the solution videos without coding along, then trying to solve it on their own. However, some students will code along with the solutions. Whichever way ensures that you will understand the material is the best way for you to go about it.

Frankeex · 14 days ago

I know this seems silly but I like to be clear and tidy and I cannot mark this lesson as completed / ticked in the side menu and I assume the % complete is wrong. Anyway we can have this done?

· Reply · Share ›

bitfountain Mod Frankeex · 10 days ago

It seems like it might be a small glitch with our website. We will look into this asap. Thanks for reporting it!

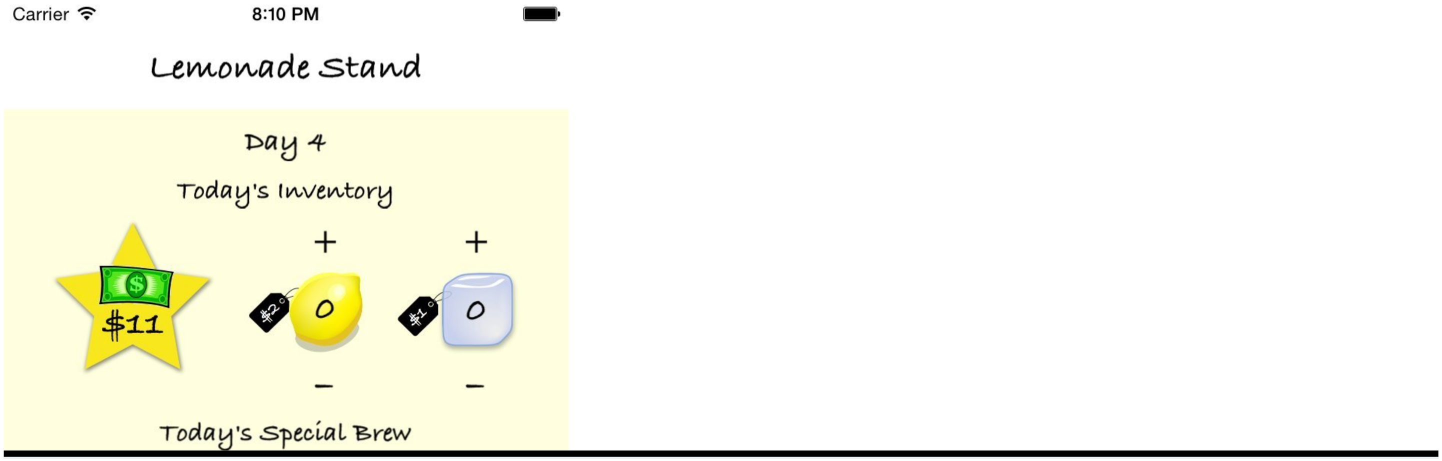
· Reply · Share ›

Vagabond Dream · 15 days ago

Here's my assignment. I made a few changes and had a bit of fun with the UI. The biggest change to the project was merging Inventory with Purchasing. Nothing 'commits' until 'Start Day' is pressed, but everything is properly constrained. The layout is fluid/responsive, so it works with all iPhone sizes. It does look a bit funny on an iPad, but it also works there as well.

https://github.com/Trundal/Lem...

I really learned a lot from this. Looking forward to the next round. :)



see more

· Reply · Share ›

bitfountain Mod Vagabond Dream · 14 days ago

Congratulations. Really nice assets. When you code something out yourself that is real it really pushes you forward as a developer. It's no small task to take a blank screen and turn it into something. I am really happy you shared this!

· Reply · Share ›

makthrow · 19 days ago

my assignment with slight variations:

disabled make lemonade button once the day's batch is made
disabled start day button if lemonade batch isn't made yet
custom UIAlert after starting day

https://github.com/makthrow/Le...

⌃ | ⌄ • Reply • Share ›

bitfountain Mod ➦ makthrow • 19 days ago

Nice job makthrow - adding your own changes is a really good step to making your own apps. Congrats!

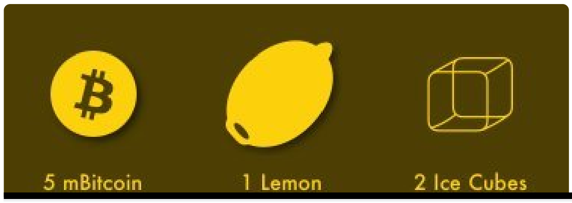
⌃ | ⌄ • Reply • Share ›

Rui • 20 days ago

Had a lot of fun with this and decided to take some time to explore some other things while working on the assignment.
The user interface is based on drag and drop instead of buttons. Except for the UIImageView used to display the weather, icons and symbols are generated using code and rendered in custom UIViews (@IBDesignable). I've used the PaintCode app for that and truly recommend it!
Some icon parameters are changeable in the interface builder (@IBInspectable).

A short video with the gameplay:
https://db.tt/QMxsUkaP
(hit download if you want to see it with a bit less blur - 11MB)

The code: https://github.com/Rygen/Lemon...
It's a bit messy :)



see more

1 ⌃ | ⌄ • Reply • Share ›

bitfountain Mod ➦ Rui • 20 days ago

Nice Job! Really Looks Great!

⌃ | ⌄ • Reply • Share ›

Rui ➦ bitfountain • 19 days ago

thanks Eliot! :)

^ | ˅ • Reply • Share ›

Vanchinathan ➦ Rui • 12 days ago

Hi Rui,
I would like to know, how did you go about learning paintcode app? Have you already worked on it? Your app looks super cool.
I would like to know, how you learnt the stuff like Present Modally to move between view controllers. I would like to understand how did you go about learning such cool tricks in such a short time. It will really help me learn some cool stuff :) THanks in advance

^ | ˅ • Reply • Share ›

Rui ➦ Vanchinathan • 11 days ago

Hello,
Thanks for your feedback!
I just read the documentation available on their website and it was enough to get around the program :)
After writing my original idea on a paper, I took some time away from the course to explore what I didn't know yet, such as drag and drop interactivity, paintcode, custom UIViews that can be rendered in the interface builder, using segues (though that is explained later in the course too),... Learnt mostly on Stack Overflow, Ray Wenderlich and some blogs I found through web searches.
Let me know if you need further details about anything.
Cheers

^ | ˅ • Reply • Share ›

Intense Wizardry • 22 days ago

Here's my completed project: https://github.com/IntenseWiza...
Thanks so much for the help with all my errors, I couldn't have done it without you. Also, I added a cool little mechanic called Ads. You can buy an ad for three dollars. Each ad lasts one day and will give you three more customers that day. You can buy up to three ads a day.

^ | ˅ • Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • 21 days ago

Cool Implementation! Keep it up!

^ | ˅ • Reply • Share ›

Intense Wizardry ➦ bitfountain • 15 days ago

https://github.com/IntenseWiza...

Here's version 2.0. I added a system where you can change the price of your lemonade.

[illegible] potentially attracting more or less customers.

^ | ⌄ • Reply • Share ›

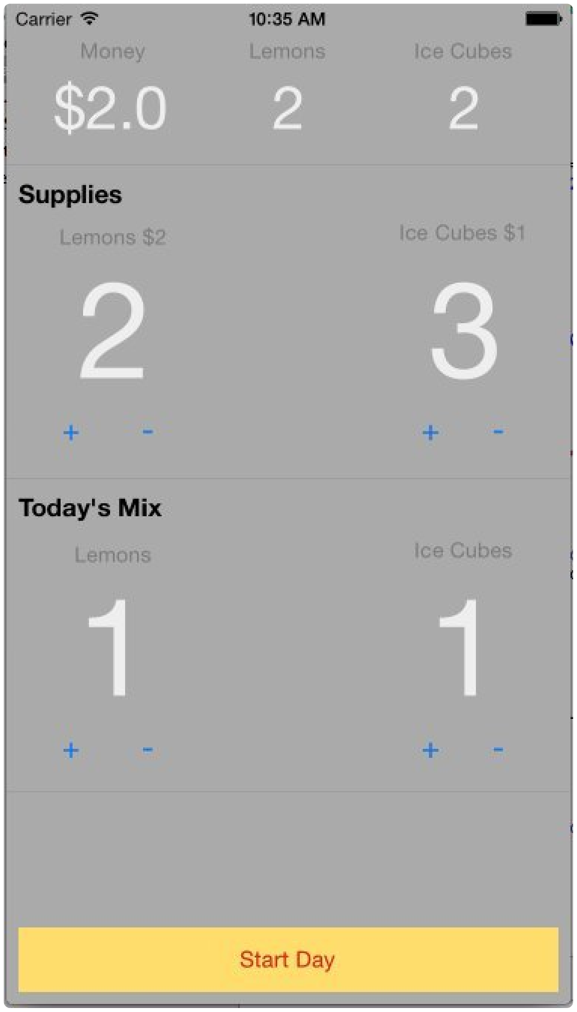
Akshay Anand ➦ Intense Wizardry • 22 days ago

hey IW, I think you might have missed adding the changes to the git change set before committing. Just go to your project directory (in terminal/console) and do a "git add ." before git commit and push commands.

^ | ⌄ • Reply • Share ›

Akshay Anand • 22 days ago

This was really fun! Thanks Eliot. Here's my solution -> https://github.com/xaksis/lemo...
and a screenshot :) Borrowed some style inspiration from Wale :)



Carrier 10:35 AM

Money	Lemons	Ice Cubes
$2.0	2	2

Supplies

Lemons $2	Ice Cubes $1
2	3
+ -	+ -

Today's Mix

Lemons	Ice Cubes
1	1
+ -	+ -

1 ^ | ⌄ • Reply • Share ›

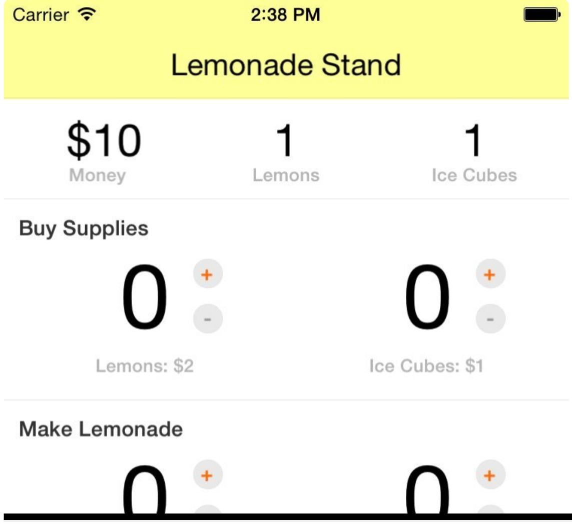
bitfountain Mod ➦ Akshay Anand • 22 days ago

Nice Job, looks great! Keep it up!

^ | ⌄ • Reply • Share ›

Wale • 25 days ago

Like everyone else has echo'd, I very much enjoyed this assignment. I'm proud of myself to be able to do this on my own and my biggest joy comes from creating the entire UI in code, without any storyboard. It's something I've been wanting to learn for a long time. Thanks Eliot! Truly a great course so far. Doing everything in code is much more work, but I see the advantages when it comes to control and customization. Now that I know both methods, I can decide when and how to use them. Can't wait to continue with the rest of the lectures.



see more

2 • Reply • Share ›

bitfountain Mod ➦ Wale • 25 days ago

So glad you enjoyed it! The final result looks great!

Reply • Share ›

Intense Wizardry • a month ago

So in assignments like this, can we go back and look previous projects? Or would you consider that cheating and discourage it? If I wasn't able to go back and look at previous projects, i would never ever be able to do these assignments probably.

Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • 25 days ago

Starting out that is perfectly acceptable. Once you become a little more comfortable writing code, we recommending struggling through errors, so that you really understand the issues that will arise. In the end, whichever way gets you to really learn the material is fine with us. Good question!

Reply • Share ›

Michael Liendo ➦ Intense Wizardry • a month ago

I would say looking back on previous projects would be perfectly acceptable. As we program and come across things we don't know, we'll have these projects to reference. The answer is never exact, but we can look at past code (especially if it's commented well) and apply it to our current project. After going back enough times, you learn what you have to do without reviewing.

1 ^ | ˅ • Reply • Share ›

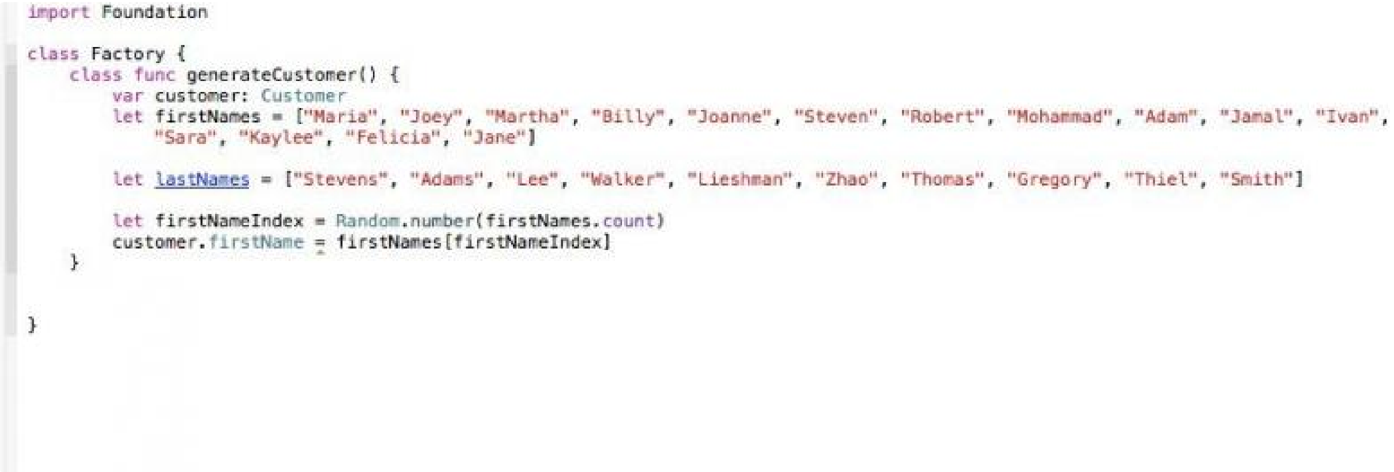
bitfountain Mod ➦ Michael Liendo • 25 days ago

Nice Post!

^ | ˅ • Reply • Share ›

Guest • a month ago

I am really really confused. So I made a file called Customer and setup a customer struct as seen below, and then I tried to make another file to generate customers. However, I keep getting this error that tells me that I can't store a value in my struct yet because it isn't fully initialized. Could someone please help?



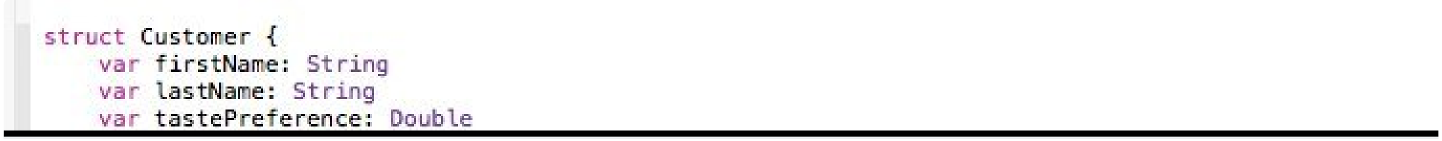
```
import Foundation

class Factory {
    class func generateCustomer() {
        var customer: Customer
        let firstNames = ["Maria", "Joey", "Martha", "Billy", "Joanne", "Steven", "Robert", "Mohammad", "Adam", "Jamal", "Ivan",
            "Sara", "Kaylee", "Felicia", "Jane"]

        let lastNames = ["Stevens", "Adams", "Lee", "Walker", "Lieshman", "Zhao", "Thomas", "Gregory", "Thiel", "Smith"]

        let firstNameIndex = Random.number(firstNames.count)
        customer.firstName = firstNames[firstNameIndex]
    }


}
```



```
struct Customer {
    var firstName: String
    var lastName: String
    var tastePreference: Double
```

see more

^ | ˅ • Reply • Share ›

Intense Wizardry ➦ Guest • a month ago

I am scouring the internet for an answer and have tried a million things, but nothing works and its really frustrating. :/

^ | ˅ • Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • a month ago

You have to add parenthesis "( )" at the end of:

var customer: Customer()

This creates an instance of customer.

^ | ⌄ • Reply • Share ›

Intense Wizardry ➦ bitfountain • a month ago

facepalm

1 ^ | ⌄ • Reply • Share ›

Michael Liendo ➦ Intense Wizardry • a month ago

happens to the best of us! I put
let iceCubesInMixLabel = formatter.stringFromNumber(IceCubeMixInt)
and kept wondering why Xcode kept saying I couldn't assign a string to a label....
spoiler: after much too long, I put iceCubesInMixLabel.text *facepalm*

^ | ⌄ • Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • a month ago

:)

^ | ⌄ • Reply • Share ›

Intense Wizardry ➦ bitfountain • 25 days ago

Another question, is it possible to create an array of UILabels? I keep trying to do that, but it never works for some reason in my code.

^ | ⌄ • Reply • Share ›

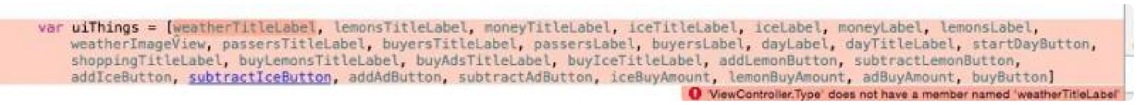
bitfountain Mod ➦ Intense Wizardry • 25 days ago

Good question! Yes, you can. Same as any other types.

^ | ⌄ • Reply • Share ›

Intense Wizardry ➦ bitfountain • 25 days ago

Whenever I try to add my array at the top with all my other variables, it gives me this error, where as if I put it in the function where it is being used, nothing happens. The reason I want it to be at the top is so that it can be a property for the whole view controller, not just a function.



```
    var uiThings = [weatherTitleLabel, lemonsTitleLabel, moneyTitleLabel, iceTitleLabel, iceLabel, moneyLabel, lemonsLabel,
        weatherImageView, passersTitleLabel, buyersTitleLabel, passersLabel, buyersLabel, dayLabel, dayTitleLabel, startDayButton,
        shoppingTitleLabel, buyLemonsTitleLabel, buyAdsTitleLabel, buyIceTitleLabel, addLemonButton, subtractLemonButton,
        addIceButton, subtractIceButton, addAdButton, subtractAdButton, iceBuyAmount, lemonBuyAmount, adBuyAmount, buyButton]
```

'ViewController.Type' does not have a member named 'weatherTitleLabel'

⌃ | ⌄ • Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • 24 days ago

Can you post your github repo link? and I'll look at your code. It worked fine for me when I tested it in my project.

⌃ | ⌄ • Reply • Share ›

Intense Wizardry ➦ bitfountain • 23 days ago

Here you go: https://github.com/IntenseWiza...

(Now I'm having a new problem. I tested my code a few times and it works, but now XCode won't compile for some reason. There aren't any errors or anything, it just refuses to compile.)

⌃ | ⌄ • Reply • Share ›

bitfountain Mod ➦ Intense Wizardry • 23 days ago

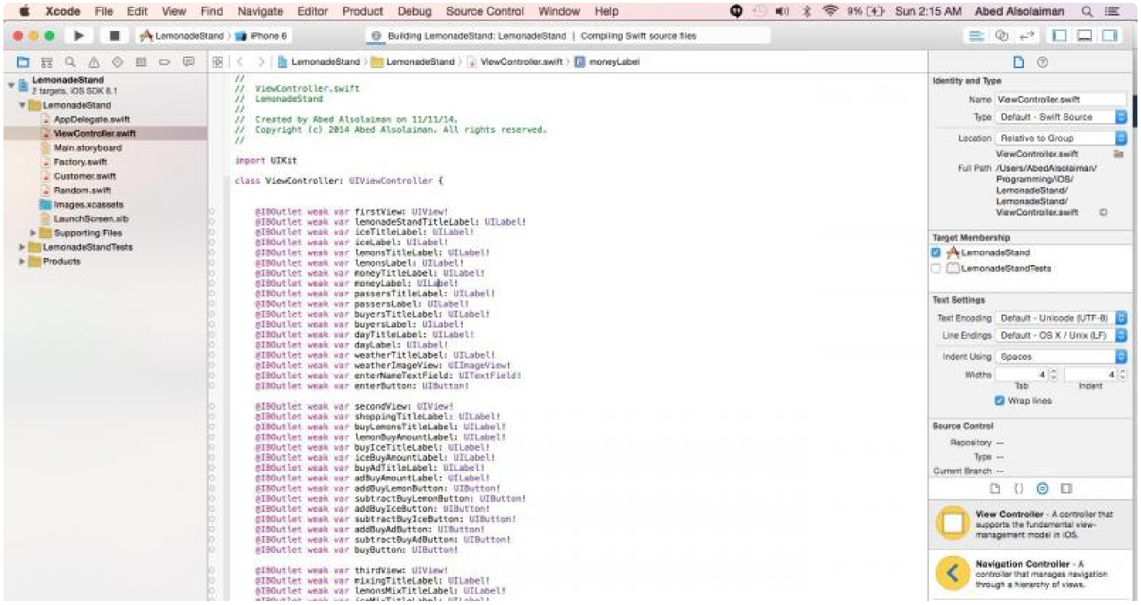
I was able to get your project to compile but I notice you have a really large number of IBOutlets perhaps this is causing an issue with your compilation. I would recommend following along with the video tutorials and then retrying this. Your setup is really complicated.

https://dl.dropboxusercontent....

⌃ | ⌄ • Reply • Share ›

Intense Wizardry ➦ Intense Wizardry • 23 days ago

Its stuck on this screen when I try to compile. (look at the bar at the top that shows the compiling progress) The same code was working before, but now it isn't.



Building LemonadeStand: LemonadeStand | Compiling Swift source files

```
//
//  ViewController.swift
//  LemonadeStand
//
//  Created by Abed Alsolaiman on 11/11/14.
//  Copyright (c) 2014 Abed Alsolaiman. All rights reserved.
//

import UIKit

class ViewController: UIViewController {


    @IBOutlet weak var firstView: UIView!
    @IBOutlet weak var lemonadeStandTitleLabel: UILabel!
    @IBOutlet weak var iceTitleLabel: UILabel!
    @IBOutlet weak var iceLabel: UILabel!
    @IBOutlet weak var lemonsTitleLabel: UILabel!
    @IBOutlet weak var lemonsLabel: UILabel!
    @IBOutlet weak var moneyTitleLabel: UILabel!
    @IBOutlet weak var moneyLabel: UILabel!
    @IBOutlet weak var passersTitleLabel: UILabel!
    @IBOutlet weak var passersLabel: UILabel!
    @IBOutlet weak var buyersTitleLabel: UILabel!
    @IBOutlet weak var buyersLabel: UILabel!
    @IBOutlet weak var dayTitleLabel: UILabel!
    @IBOutlet weak var dayLabel: UILabel!
    @IBOutlet weak var weatherTitleLabel: UILabel!
    @IBOutlet weak var weatherImageView: UIImageView!
    @IBOutlet weak var enterNameTextField: UITextField!
    @IBOutlet weak var enterButton: UIButton!

    @IBOutlet weak var secondView: UIView!
    @IBOutlet weak var shoppingTitleLabel: UILabel!
    @IBOutlet weak var buyLemonsTitleLabel: UILabel!
    @IBOutlet weak var lemonBuyAmountLabel: UILabel!
    @IBOutlet weak var buyIceTitleLabel: UILabel!
    @IBOutlet weak var iceBuyAmountLabel: UILabel!
    @IBOutlet weak var buyAdTitleLabel: UILabel!
    @IBOutlet weak var adBuyAmountLabel: UILabel!
    @IBOutlet weak var addBuyLemonButton: UIButton!
    @IBOutlet weak var subtractBuyLemonButton: UIButton!
    @IBOutlet weak var addBuyIceButton: UIButton!
    @IBOutlet weak var subtractBuyIceButton: UIButton!
    @IBOutlet weak var addBuyAdButton: UIButton!
    @IBOutlet weak var subtractBuyAdButton: UIButton!
    @IBOutlet weak var buyButton: UIButton!

    @IBOutlet weak var thirdView: UIView!
    @IBOutlet weak var mixingTitleLabel: UILabel!
    @IBOutlet weak var lemonsMixTitleLabel: UILabel!
```

Identity and Type

Name: ViewController.swift
Type: Default - Swift Source
Location: Relative to Group
ViewController.swift
Full Path: /Users/AbedAlsolaiman/Programming/iOS/LemonadeStand/LemonadeStand/ViewController.swift

Target Membership

☑ LemonadeStand
☐ LemonadeStandTests

Text Settings

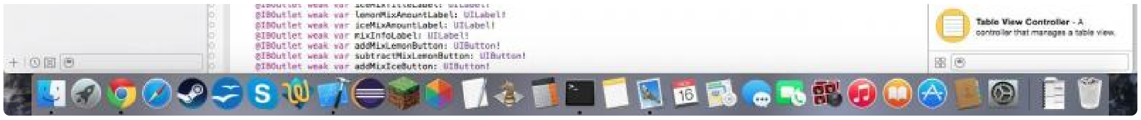
Text Encoding: Default - Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: 4 Tab, 4 Indent
☑ Wrap lines

Source Control

Repository —
Type —
Current Branch —

View Controller - A controller that supports the fundamental view-management model in iOS.

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



```
@IBOutlet weak var iceMixTitleLabel: UILabel!
@IBOutlet weak var lemonMixAmountLabel: UILabel!
@IBOutlet weak var iceMixAmountLabel: UILabel!
@IBOutlet weak var mixInfoLabel: UILabel!
@IBOutlet weak var addMixLemonButton: UIButton!
@IBOutlet weak var subtractMixLemonButton: UIButton!
@IBOutlet weak var addMixIceButton: UIButton!
```

Table View Controller - A controller that manages a table view.

^ | ⌄ • Reply • Share ›

bitfountain Mod ↪ Intense Wizardry • 24 days ago

I'm not exactly sure where you are posting it, since there is no context in the screenshot. I have to assume you are adding it above everything else. However, you still need to declare the IBOutlets for the labels above the array, if you are going to reference them in it.

^ | ⌄ • Reply • Share ›

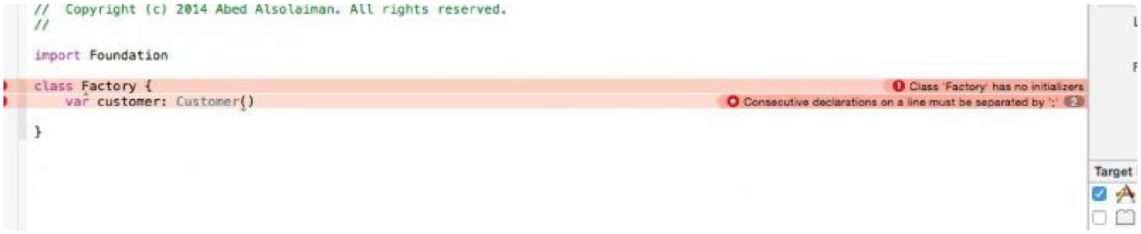
Intense Wizardry ↪ bitfountain • 24 days ago

Ah, I'm super sorry about that. I'm putting it right under all my IBOutlet declarations.

^ | ⌄ • Reply • Share ›

Intense Wizardry ↪ bitfountain • a month ago

Now it just does this:



```
//  Copyright (c) 2014 Abed Alsolaiman. All rights reserved.
//

import Foundation

class Factory {                                    Class 'Factory' has no initializers
    var customer: Customer()                       Consecutive declarations on a line must be separated by ';'  2

}
```

^ | ⌄ • Reply • Share ›

Intense Wizardry ↪ Intense Wizardry • a month ago

Nevermind, fixed it.

^ | ⌄ • Reply • Share ›

bitfountain Mod ↪ Intense Wizardry • 25 days ago

Glad you got it working!

^ | ⌄ • Reply • Share ›

Intense Wizardry • a month ago

Random question: Is there any way to add background music to our apps?

^ | ⌄ • Reply • Share ›

bitfountain Mod ↪ Intense Wizardry • a month ago

Yea, check out this tutorial link. It actually is for a game app, but there is a section in it where they show you how to add some background music, which you can implement:

they show you how to add some background music, which you can implement:
http://www.raywenderlich.com/8...

Reply · Share ›

vhristov · a month ago

Is there a way to "finish" or checkmark this lesson the way video lessons are marked as completed? Each time I'm on the home screen it says next lesson is the lemonade stand.

Reply · Share ›

Load more comments

ALSO ON **BITFOUNTAIN** WHAT'S THIS?

REV-L68-TaskIt_Updating_the_Settings_ViewController_

2 comments • 14 days ago

bitfountain — Great Point, Dominik!

Task It: Fix TableView Headers

4 comments • 15 days ago

bitfountain — No Problem!