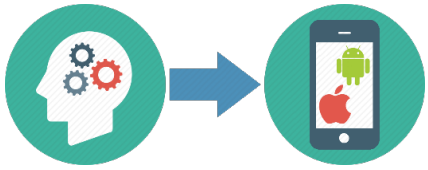
Mobile App Development

CS 4153

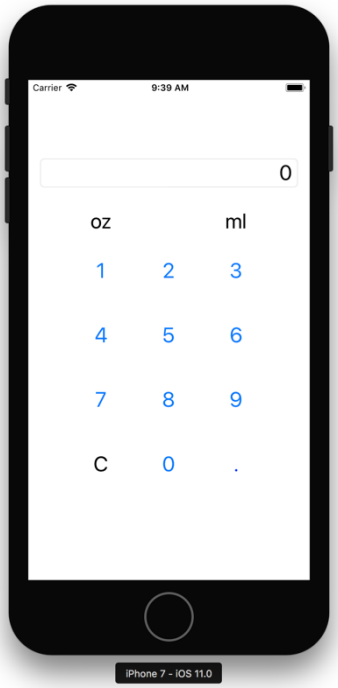


Weekly Assignment 02

Last updated: Aug 30, 2017 4:25 PM

Write an app that lets the user convert measurements between fluid ounces and milliliters.

1. Create a new, Swift 4 project in Xcode 9-beta. Name the project
   W02_ lastName_firstName
   where *lastName* is the part of your name that OSU considers to be your last name, and *firstName* is your first name. (This is the product naming convention we will use for all weekly assignments.)
2. Set the app to run on an iPhone 7.
3. When your app starts, it should look as close as possible to the image at the right. From top to bottom, it contains a text field, two buttons labeled "oz" (ounces) and "ml" (milliliters), buttons labeled with the digits 0-9, a clear ("C") button, and a decimal point (".") button. Note that the "oz", "ml", and "C" buttons are black, while the others are a contrasting color (I chose the default button color).



4. The digit and decimal point buttons are tapped by the user to enter numbers into the text field, with these restrictions:
   - Values less than 1 are the only values that can have a leading zero.
   - The value in the text field can contain only zero or one decimal points.
5. When the user taps the "C" button, reset the text field back to the value "0".
6. When the user taps the "oz" button, convert the value in the text field from milliliters to fluid ounces.
7. When the user taps the "ml" button, convert the value in the text field from fluid ounces to milliliters.

General Notes

- You can initialize the text field by entering a value in the *text* box of its attributes inspector (in the utilities area). The *alignment* menu – also in the attributes inspector – permits you to left-, right-, or center-align the text in the field.
- *Stack views* can help you arrange your buttons into nicely-spaced rows, and your rows into a nicely-spaced stack, so you may want to investigate their use. You can select several buttons and then click on the stack view icon at the bottom-right of the editor area, as highlighted in this image from RayWenderlich.com:

- Any variables you need to share among the class methods (such as those for keeping track of a running total or an array of previously-tapped values) should be declared as class *properties* (i.e., member variables of the class). As with most other OOP languages, these are declared inside the class, but outside the methods. Do not declare them as global variables.
- One way to simplify your solution is to make all of the digit buttons specify the same action method. Once inside the method, you can use the text displayed on the button to determine which button was tapped. Assuming that the button parameter for the action method is named *sender*, the text displayed on the button is stored in the property

```
sender.titleLabel.text
```

- Feel free to modify font family, font size (as I did in the image, above), font color, or the background color of any UI components. (Be tasteful in your choices.)

Submitting Your Solution

- Zip your project folder into a single file, go to the course BrightSpace site, navigate to the *Dropbox* page, and submit the zip file in the folder that corresponds to this assignment.