

SOFTWARE DEVELOPMENT UNDER IOS/SWIFT

Date: SS2021

Name: Irina Galata, Ralph Schnalzenberger

SWIFT FUNDAMENTALS

SWIFT ASSIGNMENT

Download the playground from elearning and use the following concepts from the first and second lecture:

- let/var
- Optionals
 - Chaining, guard-let, if-let, force-unwrapping, chaining
- Control flow
 - For, while, switch, etc

There is no need to hand-in this task - we will do this during the lecture and you will have some time to play around.

SWIFT ASSIGNMENT

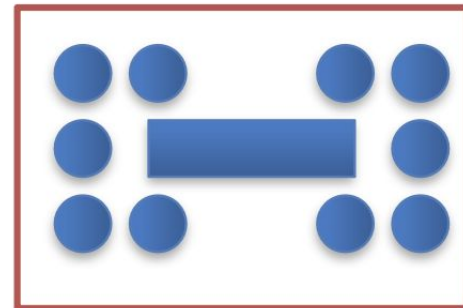
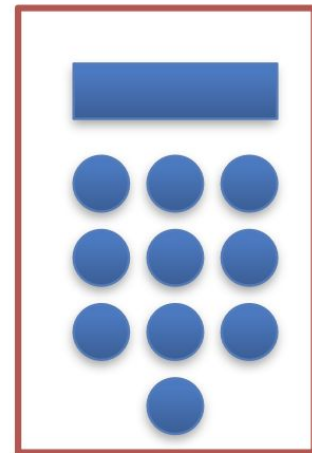
1. Implement a class **CustomInteger**
 - 1.1. This class should conform to the protocol **Equatable**. Make sure to test this by creating two different objects and check if they are equal by using `==`.
 - 1.2. This class should also use **operator overloading** to support `+` and `-`. Try it out by adding two objects and subtracting them.
 - 1.3. Look into the protocol **CustomStringConvertible** and make sure the String output of CustomInteger is beautiful.
2. Create a function that “encrypts” any given message. For “encryption” use the following code dictionary:

```
var code = [  
    "a" : "b",  
    "b" : "c",  
    "c" : "d",  
    "d" : "e",  
    "e" : "f",  
    "f" : "g",  
    "g" : "h",  
    "h" : "i",  
    "i" : "j",  
    "j" : "k",  
    "k" : "l",  
    "l" : "m",  
    "m" : "n",  
    "n" : "o",  
    "o" : "p",  
    "p" : "q",  
    "q" : "r",  
    "r" : "s",  
    "s" : "t",  
    "t" : "u",  
    "u" : "v",  
    "v" : "w",  
    "w" : "x",  
    "x" : "y",  
    "y" : "z",  
    "z" : "a"  
]
```

UIKIT

UIKIT ASSIGNMENT

- Create a simple app that allows you to enter a pin
 - Use the layout you can see on this slide
 - The Pin Field should be a label, don't use a textfield, and do not allow the keyboard usage (not the purpose of this)
 - Adapt for Size classes (using storyboards)
 - Once the app goes into background, empty the PIN field ([Hint](#))
 - Note: We will use the project for another task



DATA

DATA ASSIGNMENT

- Re-use the project from the last exercise
- Allow the user to set a PIN on the first appstart
 - Store the PIN in the UserDefaults
 - On following appstarts the user needs to enter the correct PIN
 - Either 6 pin entries (numbers) or add an enter button to confirm the pin
- Once the user entered the correct PIN
 - The user sees a ToDo list VC (TableView)
 - Empty at first (we will only store it in-memory)
 - Allow the user to add new ToDos (Button in NavBar)
 - Show a new VC with a Textfield and button

DATA ASSIGNMENT

- Update the list once a new ToDo was added on the other VC
- Allow removal of Todos via Swipe-Actions (swipe to left)
 - Hint: Override [this](#)
- Don't break any existing behaviour from before!

TESTING

TESTING ASSIGNMENT

1. Create a function which validates emails (HINT: [Regular expressions](#) can make it easier), it should return `true` if an email is valid, `false` otherwise
2. Write tests to ensure the correct implementation:
 - a. Test multiple variants of an invalid email - containing invalid characters, multiple "@" or none of it, empty string.
 - b. Test multiple variants of a valid email - uppercase, lowercase, a different language, etc.

NETWORKING

NETWORKING ASSIGNMENT

1. Get a free API key on <https://www.exchangerate-api.com> (make sure to pick FREE plan).
2. Create an app containing 2 screens:
 - a. **A list of all supported currencies names** -
<https://www.exchangerate-api.com/docs/supported-codes-endpoint>
 - b. If you tap on a currency name another **screen with conversion rates for this currency** should be shown -
<https://www.exchangerate-api.com/docs/standard-requests>
3. Use UITableView to display currencies, exchange rates and countries.
4. Show UIActivityIndicatorView to let a user know there is new data being fetched.
5. Some API's may provide an excessive amount of information, pick the fields you need to show.

PROJECT REQUIREMENTS

PROJECT REQUIREMENTS

1. Submit a link to your project link on github, bitbucket, or elsewhere.
2. Use git throughout the whole project, best would be to follow the gitflow principles.

Requirements (things your app needs to do)

- Network requests
 - Also include loading animations if applicable
- UITableView/ UICollectionView
- Min 3 distinct screens

If you want to impress:

- Store data locally (e.g. CoreData, [Realm](#), some other database technology)
- Use animations

RUNTASTIC

THANK YOU
AND HAVE A GREAT DAY



runtastic.com/career