Assignment 1

Debugger and App Life Cycle

Due: Nov. 2nd, 2015. 9:00 AM **Group size:** 2 (as assigned in the class)

Description

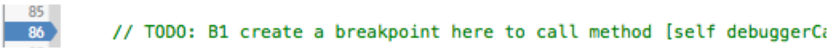
In this assignment, you will familiarize yourself with Xcode, learn how to use the debugger, and explore concepts such as app life cycle and notifications.

Task

Part 1: Xcode and debugger

1. *Hello World:* Follow the tutorial "Start Developing iOS Apps (Swift)" in the documentation. As you progress, create a list of doubts or questions that you have. This is your *personal mission list* for October–November. Review this list after each lecture or lab to check off the questions that you can answer. Raise any remaining questions that you still have in this list in the lab on November 30th.
2. *Breakpoints:* Open the `Enbugged` project. Notice that the code is written in Objective-C. Look at the syntax and try to understand the meaning of different commands. We will not use Objective-C in this course, but you will find it handy later on to be familiar with it. Search for all "`TODO:`" (Hint: ⌘⇧F, that is command + shift + F). Follow the instruction in each item to create a breakpoint or answer a question. (Hint: Try "Automatically continue after evaluating" option in the breakpoint to speed up your debugging)

   You must create a breakpoint **exactly at the same line** with `TODO:` similar to the following figure.

   ```
   // TODO: B1 create a breakpoint here to call method [self debuggerCa
   ```

   - ☐ Share the breakpoints. Submit <u>only</u> the project file: `Enbugged.xcodeproj`.
   - ☐ Use `Enbugged-answers.txt` as a template to fill in your answer. Submit this file.

Part 2: App life cycle

1. *Create an Xcode project "Clock": Use* Single View Application template.
2. *Add and link UI elements with Interface Builder:* Add a label (UILabel) to the centre of your view. Establish an outlet connection between the label and the `viewController`. In the `viewDidLoad` method, make the label display current time (Hint: use `NSDate` and `NSDateFormatter` form Cocoa Foundation framework). For debugging, add print("View Did Load") call to the `viewDidLoad` method.
3. *Using the Simulator:* Send the app to the background (⇧⌘H), then to foreground, then kill the app using the Multitasking Bar (⇧⌘H, twice quickly), force quit the app and start it again. Notice when the time label displays the correct current time and when it doesn't. How does that relate to when the debugging message appears in the console?
4. `UIViewController` *inheritance and overrideing:* Add another `UIViewController` method to your `viewController`: `override func viewWillAppear(animated: Bool) {}`. Add the

time label updating code to this method and remove it from `viewDidLoad`. For debugging, add print(“View Will Appear”). Repeat step 3.

5. *Xcode Documentation and API Reference:* Use (⇧⌘0), and read about `UIViewController` transition states and `UIApplicationDelegate` app states (the protocol implement by AppDelegate.swift).
6. `AppDelegate.swift` *transition methods:* Add break points to the methods using “Automatically continue after evaluating” option. Repeat set 3.
7. `NSNotificationCenter`*, notifications, and observers:* Read about these concepts in Xcode Documentation and API Reference or online explore: `NSNotificationCenter` and `defaultCenter` class method, and `addObserver:selector:name:object:` method.
8. Add the following code to `viewDidLoad`. Add the time label updating code to this method and remove it from `viewWillAppear`.

```
 NSNotificationCenter.defaultCenter().addObserver(self, selector:
“updateTimeLabel", name: UIApplicationWillEnterForegroundNotification,
object: nil)
```

9. Implement `updateTimeLabel` method with the time label updating code and remove it from `viewWillAppear`. Call that method from `viewWillAppear`. Run the code. Repeat step 3. The time should update now. But one problem remains: the time doesn't update continuously! (This will be tackled in the next lab.)
   - ☐ Submit `Clock.xcodeproj`

Submission

Create a zip archive including the following items

- ☐ `Enbugged.xcodeproj`
- ☐ `Enbugged-answers.txt`
- ☐ `Clock.xcodeproj`
- ☐ `Members.txt` — Modify the template to match your information
- ☐ (optional) `addendum.pdf` 1-page of anything further than the required submission

Email your submission to hamdan@cs.rwth-aachen.de with subject [iPhone 2015] A01 submission

Grading

We will grade this assignments using the following questions.

- Were the breakpoints/codes added necessary and sufficient?
- Were the answered questions demonstrate the correct and clear understanding?
- Were the implemented Clock project working?

Incomplete submission will receive at maximum 2.3.
Late submissions *will not be graded*.

Looking forward

For advanced students, the following pointers will shape your mindset for the topic we will discuss in the next lab and beyond this class.

- How to apply the MVC model to this code?
- How to make the time update continuously?
- Why do we need Deinitialization and how to use it in this project?