Assignment 1

Debugger and Memory Management

Due: October 28th, 2013. 9:00 **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 review your knowledge about memory management.

Task

Part 1: Xcode and debugger
Required knowledge: lab 1 and reading assignments from lecture 1 and lab 1.

1. *Hello World:* Follow a tutorial "Your First iOS App" 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 25th.

2. *Breakpoints:* Open the `Enbugged` project and 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.

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

Part 2: Memory management
Required knowledge: lab 1, lecture 2, and reading assignments from lecture 2.

1. *Create a project without ARC:* Create a project and disable ARC support in your project. (Hint: Google search for "disable arc project")

2. *Implement a dictionary class:* Implement a dictionary class `NSXMutableDictionary` according to the given header file. (Hint: terms such as "ownership" or "validity"are explained in document "Advanced Memory Management Programming Guide".)

   a. Your code must satisfy tests all cases specified in `NSXMutableDictionaryTest.m`

   b. You are **prohibited** from using `@property`, `NSArray`, `NSDictionary`, `NSSet`, or any of their derivatives or corresponding classes in any frameworks. You are prohibited from using convenient constructor `[NSError errorWithDomain:code:userInfo:]`.

   - ❏ Submit `NSXMutableDictionary.m`

Submission
Create a zip archive including the following items

- ❏ `Enbugged.xcodeproj`
- ❏ `Enbugged-answers.txt`
- ❏ `NSXMutableDictionary.m`
- ❏ `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 `iphone@cs.rwth-aachen.de`

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 dictionary class satisfied all given test cases?
- How well does the the code demonstrate your understanding of Cocoa memory management policy?

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.

- Describe a way that you can automatically test `NSXMutableDictionary.m` by writing a code yourself.
- How can Xcode static analyzer help testing `NSXMutableDictionary.m`?
- How can Instruments help testing your code?
- Should you migrate this code to Automatic Reference Counting (ARC) environment, what code are removed, why? Use menu “Convert to Objective-C ARC” (Hint: ⌘⇧/ allows you to search for a menu item.)
- How to modify your dictionary to work with Core Foundation objects (Hint: Documentation “Core Foundation Design Concepts”)

No submission is needed in this point, but feel free to discuss your thoughts in our Facebook group.