**CS1401 – Programming Assignment 7**

**Due: Monday October 26, 2015 by 8:00 a.m. for those of you taking the MW labs**

**Tuesday October 27, 2015 by 8:00 a.m. for those of you taking the TR labs**

**To be submitted via Piazza in the folder “lab7”**

**Welcome to Lab 7!**

|  |
| --- |
| This week, you will get more practice with arrays and Strings, which you will use within methods that you will have to design and implement.  In the last part of your lab, you will experience testing of a piece of code (one that you will have implemented earlier in lab 7), putting into practice what you have learned with the videos and in-class exercises about white-box testing. Testing is essential to your future job as a computer scientist. Pay a lot of attention to this activity! And ask questions, check with us if you are on the right track, if your testing strategy is “enough”.  We hope you have fun!  Ok, let’s get started! Here are the two activities you will be working on. |

** You should expect to work about 3 to 4 extra hours outside the lab session to complete this assignment.**

**It means that you need to make sure that Java works on your own computer, or that you go to open labs to work some more.**

**Extra time on labs includes completing the activities and taking the time to make sure that your submission is picture perfect!**

**Activity 1. Method design.**

In this activity, you will have to design and implement two methods as described below:

**Activity 1.1. Method closest**

* Takes as input: an array of integers (positive, negative, both)
* Returns two integers (as stored in an array of size 2): these two integers correspond to the two closest integers in the array
* In addition, this method displays the indices and values of these two closest integers.

Examples:

closest({-3,7,2,4,8,-19}):

* Displays something like: “The two closest integers in {-3,7,2,4,8,-19} are 7 and 8, respectively at indices 1 and 4.”
* Returns the following array: {7,8}

closest({1,-7,20,14,-18,-2,256}):

* Displays something like: “The two closest integers in {1,-7,20,14,-18,-2,256} are 1 and -2, respectively at indices 0 and 5.”
* Returns the following array: {1,-2}

**Activity 1.2. Method prefix**

* Takes as inputs: two strings, say str1 and str2
* Returns true or false as follows:
  + If str1 is a prefix of str2: prefix returns true
  + If str2 is a prefix of str1: prefix returns true
  + Otherwise: prefix returns false

Examples:

prefix(“abc”,”abcd”) returns true

prefix(“schools”,”school”) returns true

prefix(“kayak”,”kayak”) returns true

prefix(“foggy”,”bottom”) returns false

prefix(“abcd”,”abce”) returns false

**What you have to turn in:**

* In a docx file, write the pseudocode of each of the above methods: remember that a good pseudocode can be handed out to a programmer without further explanation and the programmer should be able to translate it to code. This is the standard that you will held to.
* In the java file that was provided to you, complete the description of the two methods as described above and according to your pseudocode + include in the main method the lines of code that are necessary to execute each of the above methods.
* Extra Credit: in your implementation of prefix, if you decompose your method into two method (of yours), you will receive up to 25 extra points.

**Important Notes:** 1/ If your code is not properly indented, it will not be graded (you will receive 0 for the coding part of your grade). 2/ You should not use any of the String methods such as substring(), indexOf(), etc. – you are only allowed to use length() and charAt(…).

**Activity 2. White-box Testing.**

In this activity, you are expected to design and describe a white-box testing strategy for method closest completed in Activity 1.1. Make sure you are convincing when you justify to us (instructors and TAs) that your testing strategy is thorough.

**What you have to turn in:**

* In the same docx file as in Activity 1, describe and motivate your testing strategy.

**Important notes:**

* Indent your code properly following guidelines available at: <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-136091.html>. Badly-indented code will be graded 0.
* Spend time working on your pseudocode as the amount of points you get for the pseudocode is bigger than the amount of points you get for your code (usually, close to a 60/40 ratio).
* Do not submit more than the files that are requested from you: one docx file and one java file.

**That’s it! Looking forward to seeing you in lab!**