CSC220 Assignment 1 – Java Review

For your first assignment, imagine that you are given an array of integer values. The values in the array are sorted in an ascending order and you want to figure out whether there exists any pair of numbers (i.e., any two elements) in this sorted array that will add up to 20. You need to write a simple function that perform this task. For simplicity, you can assume if there exists such as pair, it is unique. Please follow the following steps exactly to get started:

1. In your lab01 package, create a class (or Java file) as you learned and call it: **SumExperiment.java** (nothing else!).

2. Log into Blackboard and copy the provided **SumExperiment.java** file into your src folder (or copy the content in the file you just created). You should now see a **main** function and another function called **checkSum**.

3. As you can see the **checkSum** function, receives an array of integers as its input and returns an integer value. The return value is the **index** for the smaller value in the pair. If such a pair does not exist in the array, the function is supposed to return -1. For instance, if we give the following array to **checkSum** function: {5, 7, 8, 9, 10, 15, 16}, **checkSum** should return 0. Why?

4. IMPORTANT NOTE: DO NOT CHANGE THE SIGNATURE OF THE FUNCTION OR THE JAVA FILE. IF YOU CHANGE THE SIGNATURE OF THE FUNCTION (ITS NAME, INPUT OR OUTPUT OR THE FILE NAME) YOU WILL LOSE POINTS (rational: we use automated scripts for grading so if you change things, the grading will become tedious for the TA who is a student just like yourself!)

5. Now start thinking about how would you accomplish this task. Take a moment to come up with a solution on your own before you go to the next step!

6. The naïve way to accomplish this task is to check all the pairs in the array. However, remember that the array is sorted, can we somehow deploy this property do this task more quickly? Again, try to think about it before you look at the next step. Hint: think about how you can use two pointers (or iterators) to accomplish this task more

efficiently than looking at all pairs.

7. Okay, hopefully you already got the idea. If not, I strongly encourage you to stop reading and think about it – I will NOT provide the idea every time ;)

8. Here is how you can do it more efficiently: Imagine you have two pointers (they can simply be two integer values that keep the index of the array), one pointing to the beginning of the array and advances forward and the other one points to the last element of the array and advances backward. All you need to do is to sum the values these two pointers point to and see if they are equal to 20, if so, great! you can prepare the output of the function.

9. If not, you inspect the sum, if the sum is greater than 20, you will decrement the second pointer and if the sum is less than 20, you increment the first pointer.

10. Before start coding, try the idea on a piece of paper with some example (you can use the ones provided in the main function) and convince yourself that this would work.

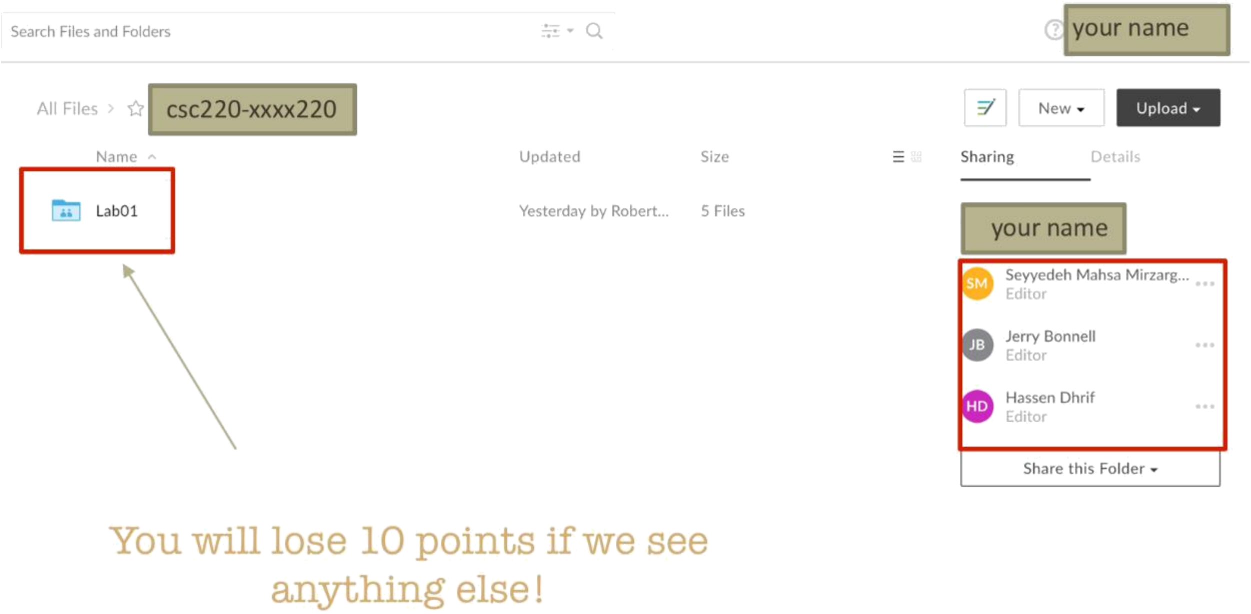
11. Now, start writing the body of the function. Again, do not change the signature!

12. After you are finished writing the function, you can use the main function provided to test your code. You see that we have four arrays in there. Try to work out each case in your mind (or with pen and paper) and understand what those "if" statements are going to do.

13. If you see any "TEST FAIL" when you run your program, something is wrong with your function. Go back and debug your function. If your function is working properly, you show only see "Done!!!!". If you see anything else, something is wrong!

IMPORTANT NOTE: You are required to use the two-iterators idea, if you implement the naïve approach or any other algorithm you will lose points.

Here is how your box folder should look like after submission of your first assignment – if it looks any different please consult the submission guideline again (in tutorial folder in "course document section of Blackboard).



You will lose 10 points if we see anything else!

For all your assignments, please start early and seek help early (either from the instructor or the TAs).

Make sure to follow the submission guidelines you learned during the first lab (also available in the tutorial folder in the "course document" section of Blackboard).