EECS1022 Winter 2018
Guide to Lab Test 2

Chen-Wei Wang

In this lab test you will be required to code Java methods in Android Studio, using **arrays and loops**. You need not worry about declaring any classes, attributes, or methods. A class declared with methods will be given to you, and you only need to worry about writing lines of code that serve as bodies of implementations of these methods.

1 Coverage

– All lecture materials (slides, recordings, and example codes) covered up to and including Monday, February 26 (in particular loops and arrays).

– Lab 3 and Lab 4 are not needed for this lab test.

2 Solutions

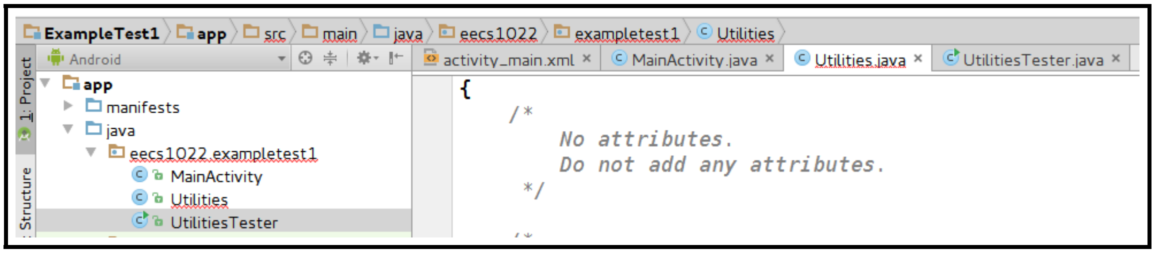
Video solutions to the **example test** will be made available during the reading week. You are advised to first attempt all problems.

3 Rules

– This lab test is **purely** a programming test:

- You are forbidden to use Java library classes and methods, e.g., `Math`, `Arrays.sort`, *etc.*.
- You are given starter code (i.e., `Utilities` and `UtilitiesTester`) that compile. See the preparation exercise below.
- As you fill in bodies of implementations of methods in the `Utilities` class, you must not introduce any compile-time syntax or type errors.

  An advice for you is to pay close attention to the tags of these two classes, as soon as you see any of these two tags has a red underline, e.g.,



```
{
    /*
        No attributes.
        Do not add any attributes.
     */
```

  then your priority is not to continue your development of code, but to fix the compile-time errors.

- You will only need to submit the `Utilities.java` file:

  ∗ If this class does not compile due to any syntax or type errors, you receive **zero** marks for the test. There will be no partial marks rewarded to a class that does not even compile.

This requirement will be imposed strictly, so your best preparation strategy is to familiarize yourself with all basic syntax of Java that's so far covered in the lectures and used in your labs.

  - ∗ If your submitted class compiles, then you already receive **10 marks** (out of 100) of the test.
  - ∗ To determine the remaining 90% of your marks, we will run a number of test cases on your submitted class.

    For example, say we run 10 test cases (and say they happen to have equal weights) on your submitted code, and your submitted code complies and passes 6 of them, then your final marks are: $10 + 80 \times \frac{6}{10} = 58$.

- You will be given **80 minutes** for the lab test.

- You must show up for your registered session only.

- Bring a piece of photo ID.

- No mobile phone usage is allowed during the test.

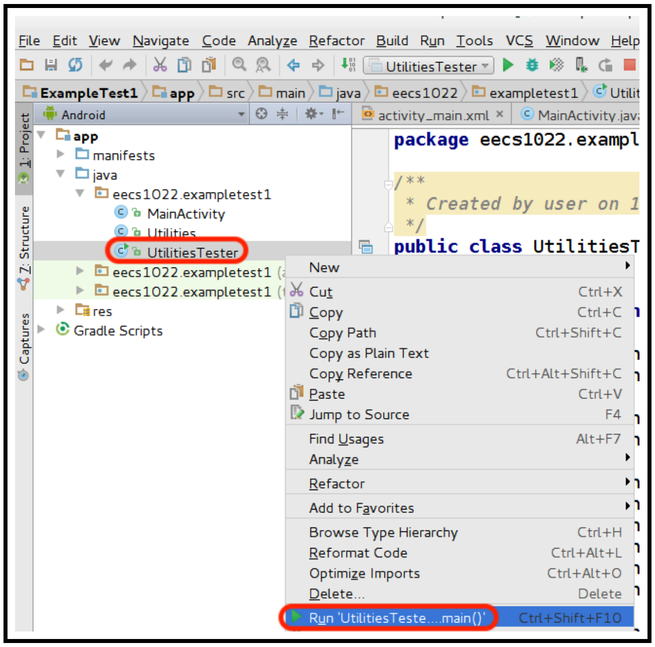
- No data sheet will be allowed.

- You may bring pen/pencil and a piece of blank paper for sketching your solutions.

4 Format

- The format will be similar to the **preparation exercises here** (click on the link to download), where you are be given:

  - A class `Utilities` with a list of methods, each of which with its body of implementation returning something useless. You are forbidden to change signatures of methods.
  - A class `UtilitiesTester` which demonstrates a list of use cases of methods of the `Utilities` class. To execute this tester class, in your Android Studio, go to the `Project` panel on the left, go under `app/java/eecs1022.exampletest2`, then right-click on `UtilitiesTester` and select `Run 'UtilitiesTester.main()'`.



```
package eecs1022.exampl

/**
 * Created by user on 1
 */
public class UtilitiesT
```

Running `UtilitiesTester` will execute methods in the `Utilities` class and display output on the console in Android Studio. No physical connection to a tablet will be necessary in the test.

- For the example test you're given, here's the expected output (which is not the case until you correctly implement all methods in `Utilities`):

```
============averageOf
75.0
76.0
15.0
14.8
============allMultiplesOf5
true
true
false
true
false
============atLeastOneMultipleOf5
false
true
false
true
true
============secondMaximumOf
5
5
9
10
9
============reverseOf
{}
{75}
{76}
{25,20,15,10,5}
{25,19,15,10,5}
============isReverseOfEachOther
false
false
false
true
true
============getArithSeq
{}
{3}
{3,5,7,9,11}
{}
{3}
{3,1,-1,-3,-5}
============isArithSeq
true
true
true
true
true
true
false
false
```

```
============getFibSeq
{}
{1}
{1,1}
{1,1,2}
{1,1,2,3}
{1,1,2,3,5}
{1,1,2,3,5,8}
{1,1,2,3,5,8,13}
============isFibSeq
true
true
true
true
true
false
false
false
false
============numberOfOccurrences
4
1
2
3
0
0
0
============getIndices
{2,6,7,9}
{3}
{1,4}
{0,5,8}
{}
{}
{}
```

- Your tasks are to fill in, for each method in the `Utilities` class, lines of code in valid Java syntax as its body of implementation, such that running `UtlitiesTester.main()` will produce the identical output as expected.