Software Testing

Assignment 1

In this assignment, you will practice activities covering the concepts taken in the course so far, through two main tasks as follows:

Task 1

Below is one faulty program. It includes test inputs that result in failure. Answer the following questions about this program.

Program:

```
/**
 * Compute an average
 *
 * @param arr array of numbers
 * @return average of numbers in arr
 * @throws NullPointerException if arr is null
 */
line  1  public static double computeAverage(double[] arr)
line  2  {
line  3     if (arr == null)
line  4        throw new NullPointerException();
line  5
line  6     double sum = 0.0;
line  7     for (int i=0; i < arr.length-1; i++)
line  8        sum += arr[i];
line  9
line 10     double average = sum / arr.length;
line 11     return average;
line 12  }
```

a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.
b) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not.
c) If possible, give a test case that results in an error, but not a failure. If not, briefly explain why not. Hint: Don't forget about the program counter.

d) if possible, give a test case that results in a failure. If not, briefly explain why not. Note: for credit, your test case must not be the same as the given test case.

e) For the given test case

```
// test input: arr = [90.5, -65.0, 72.25]
// expected = 32.58
```

Describe the first error state. Be sure to describe the complete state.

Task 1 deliverables:

1. A report showing a detailed answer for each step for the **above function** from (a) to (e). This report can be a word document, or a set of compressed scanned pictures.

Task 2

Objectives

- Studying implementation of an open source project.
- Using Input space-partitioning & boundary-analysis techniques to design test cases.
- Learn unit testing and code coverage.

Description

- In this task, you will study the open source code of "GeoProject" **attached**.

geohash is a convenient way of expressing a location (anywhere in the world) using a short alphanumeric string, with greater precision obtained with longer strings.

- Docmentaion of geo classes check https://davidmoten.github.io/geo/apidocs/index.html
- Geohash Demo https://www.movable-type.co.uk/scripts/geohash.html
- You are to study all 6 classes of the SUT (GeoProject)
- Once you study those APIs implementations/documentations, you are required to use input space partitioning along with boundary values analysis to design unit tests for all methods in **GeoHash.Java** class using **input space-partitioning**.

Task 2 deliverables:

2. A report of all the test cases you have developed for the classes mentioned above, with full reasoning of the partitions, the boundary values, and the chosen coverage criteria with complete explanation. In the written report, you should discuss how you are designing the test cases. This report can be a word document, or a set of compressed scanned pictures.
3. A testing package, with a test class for each class mentioned above (please follow java coding/naming standards).

Things to consider for task 2

• You can use JUnit or TestNG or any other java unit testing library if you please.

• Failing reasons for each unit test case should specifically be one reason.

• You should add your test package/classes to the included project.
• Make the test design document as detailed as possible.

General Assignment Guidelines

- This assignment is in groups of two **at most**.
Team members can belong to different Groups if the 2 groups share the same TA.
- All the deliverables from both tasks should be placed in one folder, and should be named as stud1ID-stud2ID-Assignment1.zip
- Failing to name the zip file as mentioned above, marks will be deducted. -
In case of cheating, a faculty-based regulation will be applied

Due date and submission

Assignment 1 is due on Sunday, May 6th at 11:55 PM (Cairo Local time). Submission needs to be done through the course's Blackboard only.

Late submission policy: Late submissions is allowed with a penalty for two days only. A 50% late submission penalty will be applied after the original deadline.