Homework 2

Note:: The questions in "Test Automation Mindset" should be answer individually. There are not right or wrong answers to this section, only complete and incomplete.

1 Test Automation Mindset

1.1 Limitations

What are the limitations of test automation? Do you forsee a way to overcome any of the limitations you listed?

1.2 Testing Tool Creation

You find yourself in the position to build your own test automation toolset. Give at least 4 features that you would ensure your tool is able to provide. For each feature, explain why it is needed.

Note: You may collaborate on the remainder of the assignment. If you do collaborate on the assignment, list your collaborators. All duplicate assignments without collaborators listed will be flagged for plagarism. Additionally, presenting others work – including a Chegg expert answer – as your own work *is* a violation of the academic dishonesty policy.

This homework assignment is designed to prepare you for the in class quiz.

2 Explore a Test Suite

In the homework zip file, there is a `Factors.java` that we will explore in this section. There are 4 methods in `Factors`: one implementation of `aFactor` and 3 different versions of `aNonTrivialFactor` with the same pre-/post-conditions but different implementations.

Your job is to execute the existing test suite and:

(a) For each test case provide the following:

- The outcome of running the test case: pass, error, or failure
- If there is any defect exposed (or hidden) by the test case, specify where the problem is, e.g., program under test or test case code and what kind of problem you discovered
- Fix the problem in the method under test or in the test and briefly describe here how you resolved it.

(b) Come up with at least 3 significantly different new test cases for one of your corrected implementations of `aNonTrivialFactor`.

(c) Submit a corrected version of `Factors.java` such that all your tests pass.

3 The Classic Account Class

In the homework zip file, there is an `Account.java` class which is a tried-and-true classic example class that new computer science students often have to write.

Your job is to:

(a) Write a JUnit test suite such that all the methods of the `Account` class are explored in at least one test. Keep your JUnit test suite in `AccountTest.java`.

(b) If your tests reveal any defects in the `Account` class, repair the class and submit `AccountRepair.java` with comments to highlight your changes.

(c) There is a subtle bug in `Account` that your tests may not have revealed. Examine the same code in `UnwantedBehavior.java` and make sure you have a test case that reveals this bug in `Account`.

(d) How would you correct the bug in (c)? Give a written explanation, do not attempt to repair the problem.