

Homework 1

Note:: The questions in “Current Testing Mindset” should be answer individually. There are not right or wrong answers to this section, only complete and incomplete.

1 Current Testing Mindset

1.1 Testing Roles

Answer question (a) or (b), but not both, depending on your background.

- (a) If you do, or have, worked for a software development company, how much effort did your testing / QA team put into each of the four test activities? (test design, automation, execution, evaluation)
- (b) If you have never worked for a software development company, which of the four test activities do you think you are best qualified for? (test design, automation, execution, evaluation)

1.2 Testing Philosophy

“Complete testing is impossible and therefore testing involves tradeoffs.”

You find yourself in charge of Google’s search engine. You have a new algorithm to improve on the existing Google’ search result rankings. The code base you are testing is *live* meaning that end users will experience your software update *immediately*, although in a rolling basis where gradually more users will be exposed to the new search engine. What would be your testing plan for the rollout?

Feel free to describe more details of a testing plan, but make sure to answer the following:

- (a) Who would do the testing?
- (b) How would you define “correct” behavior?
- (c) Describe how you’d set up one illustrative test case. Tell me what the input, expected output, and overall test environment is.
- (d) What would be your condition to stop testing?

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 plagiarism. 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 Summing natural numbers

```
/*  
 * Find the sum of the first "n" natural numbers  
 * @param integer n  
 * @return the sum from 1 to n  
 * @throws error if n is not a valid natural number  
 */  
public int findSum (int n) {  
    int sum = 0;  
    for (int i = 1; i < n; i++) {  
        sum += i;  
    }  
    return sum;  
}
```

Based on the code above, answer the following questions:

1. Give a sentence or two explaining what is wrong with the code. Include a suggested fix that would produce a corrected version of the program.
2. If possible, identify a test case that does **not** execute the fault. If not possible, explain why.
3. If possible, identify a test case that executes the fault, but does **not** result in an error. If not possible, explain why.
4. If possible, identify a test case that results in an error, but **not** a failure. If not possible, explain why.
5. If possible, identify a test case that results in a failure.
6. Identify the first error state for the given test from (4) or (5). Indicate which test case you are using.

Now, let's take a look at a two different scenarios:

- (a) The sum of the first n natural numbers can be captured by the formula:

$$\sum_{i=1}^n = \frac{n(n+1)}{2}$$

Assume you have a method that uses this equation called `sumShortcut`. If you are trying to systematically generate tests to verify `findSum`, would there be a test you'd generate for `findSum` but not `sumShortcut`? (In other words, would the presence of a loop in one but not the other impact your test case design when thinking about trying to propagate errors?) Explain.

- (b) The sum of the first n natural numbers is sometimes considered as the sum from 0 to $n - 1$, as sometimes people count 0 as a natural number. If this is the expected behavior, would any of your answers to 1 - 5 change? If so, give the new answers, if not state why.

3 Sum of digits

```
/*
 * Find the sum of the digits that make up a number
 * @param integer n
 * @return the sum of all the digits so 323 should return 8
 * @throws error if n is null
 */
public int sumDigits (int n) {
    int sum = 0;
    while (n != 0) {
        // Add the last digit to the sum
        sum = n % 10;

        // Remove the last digit
        n = n / 10;
    }
    return sum;
}
```

Based on the code above, answer the following questions:

1. Give a sentence or two explaining what is wrong with the code. Include a suggested fix that would produce a corrected version of the program.
2. If possible, identify a test case that does **not** execute the fault. If not possible, explain why.
3. If possible, identify a test case that executes the fault, but does **not** result in an error. If not possible, explain why.
4. If possible, identify a test case that results in an error, but **not** a failure. If not possible, state explain why.
5. If possible, identify a test case that results in a failure.
6. Identify the first error state for the given test from (4) or (5). Indicate which test case you are using.

A single line implementation is given below and contains the same general fault:

```
public int sumDigitsOneLine (int n) {
    int sum;
    for (sum = 0; n != 0; sum = n % 10, n /= 10);
    return sum;
}
```

Let's consider the following situations:

- (a) If you are trying to systematically generate tests to try and propagate errors in `sumDigits` and `sumDigitsOneLine`, would there be a test you'd generate for one but not the other? Explain.
- (b) Correct the fault in `sumDigitsOneLine`. Would any of your answers to 1 - 5 from `sumDigits` change? If so, give the new answers, if not state why.