5321 Homework 4

Summer 2020

Question Weighting:

Question 1-5 - 20 percent each, question 6 is worth an extra 20 bonus points

**Files provided to you.**

Java source files - this is the code you developed test cases to in HW 3

1. Each Java file is attached in the Canvas zip file. Please use these files. DO NOT modify the code - if you have to modify the code there is something wrong with your test.

Excel Test Cases Tables

1. These are the test case tables from HW 3. You will save as a csv file to use with JUnit

**Submittal items**

Submit the following in the PDF file for each problem

1. Test case table snapshot
2. JUnit pass indicator (green bar expanded to see values where possible)
3. JaCoCo statement green source line annotations (not JaCoCo summary). Make sure to include the time stamp on each JaCoCo screen shot.

Include in ZIP file

1. Your homework solution as a pdf or Word (previous item)
2. JUnit test files (make sure problem number is referenced in the file name)
3. csv files used (make sure problem number is referenced in the file name)

Problem 1

Use the code from Canvas and the provided test cases we developed from Homework 3. Implement this using the **JUnitParamsRunner**. You will need to declare the expected motorState and chuteState variables in the JUnit test method as the following:



and in the JUnit test case table enumeration constants are referenced as the following:

Problem1Class.motorStates.***Off***

Problem 2

Use the code from Canvas and the test cases we developed from Homework 3. Implement this using the **FileParameters** (read the values from a file).

Problem 3

Use the code from Canvas and the provided test cases from Homework 3. Implement this using the **FileParameters** (read the values from a file).

Problem 4

Use the code from Canvas and the provided test cases from Homework 3. Implement this using the **FileParameters** (read the values from a file).

You will declare the expected return value (enumeration) in the JUnit test method as in Problem1.

Problem 5

Use the code from Canvas and the provided test cases from Homework 3. Implement this using the **FileParameters** (read the values from a file). For the expected value of y use a comparison threshold of **0.001** for assertEquals.

Problem 6 - **20 extra bonus points**

Use the code from Canvas and determine what it does. Develop a logical expression for each of the values produced (1-5). Use MC/DC to determine the test cases for each value produced. When developing the MC/DC test cases, if a condition is not present in the logical expression use a false (F) for that condition in each MC/DC developed test case. For example, if developing the test cases for the logical expression abc then solve MC/DC for abc and use the value of false (F) for condition d in each test.

Q1 (10 points): Of the developed MC/DC tests give the logical expression for the faults that cannot be detected.

Q2 (10 points): For each of the terms in this logical expression indicate the range of possible erroneous values produced. For example abc (2-4).

No hints, explanations, or partial credit will be given. You must answer each part exactly to get the points indicated.

Solution:

Logical expression for each of values produced:

Value 1: abcd’

MC/DC: TTTF, FTTF, TFTF, TTFF, TTTT

Value 2: abcd

MC/DC: TTTT, TFFF, FTFF, FFTF, FFFT

Value 3: abc’ + ab’c + ab’c’

MC/DC: FTFF,TTFF,TFFF,TTTF

Value 4: a’bc’+a’b’c’

MC/DC: TFFF,FTFF,FFFF

Value 5: a’b’c

MC/DC: FFTF, TFTF, FTTF, FFFF

***Question 1:***

Logical expression for Value 1: abcd’

logical expression for the faults that cannot be detected:

Logical expression for Value 2: abcd

logical expression for the faults that cannot be detected:

Logical expression for Value 3: abc’ + ab’c + ab’c’

logical expression for the faults that cannot be detected:

Logical expression for Value 4: a’bc’+a’b’c’

logical expression for the faults that cannot be detected:

Logical expression for Value 5: a’b’c

logical expression for the faults that cannot be detected:

***Question 2:***

Logical expression for Value 1: abcd’

Range of possible erroneous values: 2-3(2,3)

Logical expression for Value 2: abcd

Range of possible erroneous values: 3-5 (3,4,5)

Logical expression for Value 3: abc’ + ab’c + ab’c’

Range of possible erroneous values: 4

Logical expression for Value 4: a’bc’+a’b’c’

Range of possible erroneous values: 3,5

Logical expression for Value 5: a’b’c

Range of possible erroneous values: 3-4(3,5)

**Submission Evidence (numbers below correspond to numbers above in the assignment section).**





1. **Snapshot for JaCoCo coverage report ( HW 4 and HW 5) must include the timestamp of the local machine**
2. Submit your HW 4 as a single zip file in accordance with the class HW submission standards