



Name:

Vnumber:

## **SENG 275 MID TERM EXAM 2(10%) Summer 2023-O5**

Instructor: Dr. Navneet Kaur Popli, Date: 27 June 2023, Time: 12:30-1:20 PM PST

Mode: Pen-Paper, Synchronous, Timed, to be done in the question paper, closed-book,

Total Marks: 50, Total questions: 16

**Note:** This is an individual activity. Copying or cheating of any kind is not allowed. Make a note of any assumptions you make.

Q1) Purchase discount is 0% for up to 500 US \$ purchases, 5% is added for each additional 500 US \$ up to 2000 US \$, and 25% is applied for above 2000 US \$. Which test inputs in US \$ would be selected for valid equivalence partitions? Select one option. (1M)

- a) 250, 700, 1400, 1800, 4000
- b) 250, 1400, 3000
- c) -100, 250, 650, 1300, 1700, 2900
- d) 200, 720, 1600, 1800, 2100

Q2) Regarding **boundary analysis of the condition**  $a \leq 10$ , which of the following statements **is true (for this condition)**? Select one option. (1M)

- a) There can only be a single on-point which always makes the condition true.
- b) There can be multiple on-points for a given condition which may or may not make the condition true.
- c) There can only be a single off-point which may or may not make the condition false.
- d) There can be multiple off-points for a given condition which always make the condition false.

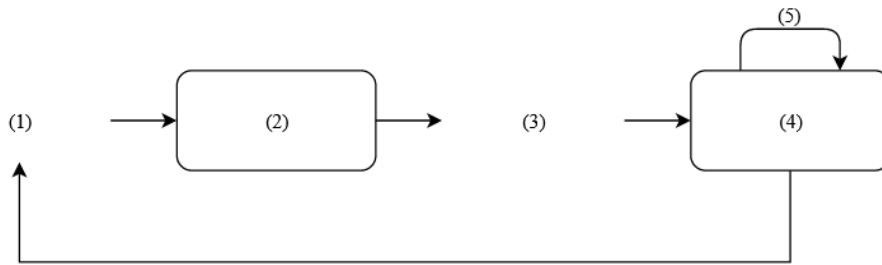
Q3) We want to perform boundary analysis on the following equality:  $x = 10$ . Give one on and one off point? (1M)

A3)

Q4) It is possible for a test to fully cover a line, without achieving full branch coverage of that line. Select each of the following types of code where this phenomenon could occur: (1M)

- Conditionals
- function calls
- variable declarations
- FOR loops and enhanced FOR loops (foreach loops)
- object calls

Q5) We have the following skeleton for a diagram illustrating the Test-Driven Development cycle. What words/sentences should be at the numbers? (2.5M)



Q6) Given below is a feature and a user story from a shopping cart website. Write Given, When, Then, And (And can be use multiple times if required) BDD test scenarios for this feature. Write one valid and one invalid scenario (an invalid scenario may not depict a buggy behaviour in a well tested and well-developed application). Note any assumptions that you make. (2+2=4)

Feature: Adding items from wish list to shopping cart.

User Story: As a consumer of a shopping website ABC.com, I should be able to add an item from wish list to shopping cart.

A6)

Q7) Give any 3 advantages of following the BDD and TDD approach. (3M)  
A7)

Q8) Request Traceability Matrix is a document that traces..... with  
..... (2M)

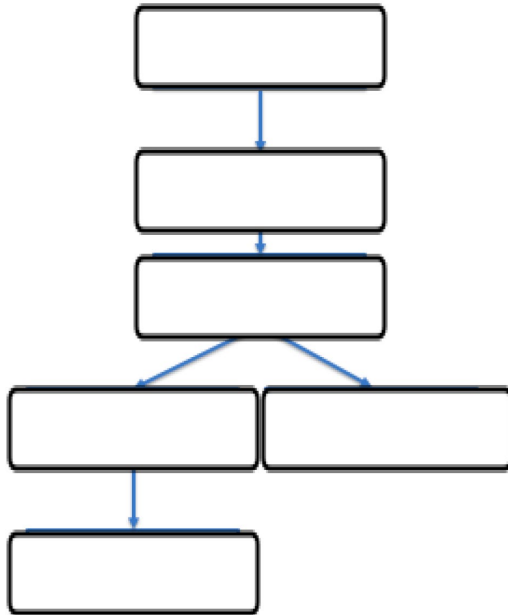
Q9) Assume that UVic's bug tracking software has two metrics associated with each bug - "Bug Priority", and "Bug Severity". Explain the difference between these two measures using any example from UVic's Brightspace page. (2M)  
A9)

Q10) Explain (using a diagram) the Bug resolution matrix. (2M)  
A10)

Q11) JUnit randomizes the order of the unit tests it runs, each time the overall suite is executed.  
Why is this desirable? (1.5M)  
A11)

Q12) Some strategies subsume other strategies. Formally, a strategy X subsumes strategy Y if all elements that Y exercises are also exercised by X. For the following six coverage strategies, fill in the hierarchy below to show the subsumption relationships among these strategies: (3M)

1. Branch + Condition coverage
2. MC/DC coverage
3. Path coverage
4. Branch Coverage
5. Block/Line coverage
6. Condition coverage



Q13)

If ( (( person == student) || (person == professor)) && ((department == ECE) || (department == CS)) )

{

    return ("You are invited to the Software conference");

}

else

{

    return ("You are not invited to the software conference");

}

Use the MC/DC approach of software testing to create minimum number of test cases to test this piece of code. Explain the steps you take. (10+2=12M)

A13)

Q14) Write a parameterized test using `@ValueSource` or `@CSVSource` to test the function `compassDirection()`, which returns a string containing the closest cardinal direction (North, East, South or West) as a string, given an integer number of degrees (with 0 degrees corresponding to due north). Tight boundary testing is not necessary, but you should have at least one test for each possible output and remember to check negative or otherwise potentially problematic integers. The code for the function is shown below: (5M)

```
public static String compassDirection(int degrees) {
    if (degrees < 0) degrees -= degrees; /*this is degrees+=360. The idea is
to coerce degrees to [0,359] */
    degrees %= 360;
    if (degrees > 315 || degrees <= 45) return "North";
    if (degrees <= 135) return "East";
    if (degrees <= 225) return "South";
    return "West";
}
```

A14)

Q15) Using the TDD approach, write one 'Before' annotation code, and two Tests for a Rectangle class: a Before annotation has 'createObject()' function which creates a new rectangle with length and breadth as integers. The createObject() should have parameters for length and breadth. Also, its return type should be Rectangle. This object can be used by the next two tests. Create code to pass the test of Before annotation. Then create a test for the 'area()' function which calculates area of the rectangle as length\*breadth. Create code to pass this test. Then create a test for the 'perimeter()' function which calculates the perimeter of the rectangle as 2\*(length+breadth). Create code to pass this test. Do not repeat code in subsequent tests. Write this in a step-by-step form in a way TDD should be performed. Suggest any one way in which the code can be refactored. Write the refactored part of the code. Do not add any new functionality in the refactoring step. (2+2+2+2=8M)

A15)



Q16) Is code coverage a good measure of test suite effectiveness? Comment. (1M)  
A16)

END