

SWT301 - FALL23 - The final PE

Duration 90 minutes

Instructions:

- You need to know at least one programming language to perform reasonably well on common operations.
- Students are allowed to use IDEs such as NetBeans, IntelliJ IDEA, or similar software to write their code.

Question 1 (10 points) You are assigned to do the following class and test for basic 8 issues in the code (i.e., coding practice, except errors, potential logical errors, etc.). (For practice / template)

```
public class Fibonacci {
    public static long calculateFibonacci(int n) {
        if (n <= 0) {
            System.out.println("The input number is invalid.");
            return 0;
        }
        if (n == 1) {
            System.out.println("The result is 1.");
            return 1;
        }
        long result = 0;
        long first = 0;
        long second = 1;
        for (int i = 2; i <= n; i++) {
            result = first + second;
            first = second;
            second = result;
        }
        System.out.println("The result is " + result);
        return result;
    }
}
```

Question 1 (10 points) Assuming you are assigned to conduct the component test for the method below, please design and create the minimum requirement test cases (Unit Test) and modify the code to achieve 100% statement coverage and 100% decision coverage. (The question 2 complex).

This code defines a *calculator* provides method for value has modulus and an operator *divide* as input. It performs the specified operation (addition, subtraction, multiplication, or division) and returns the result. It includes logging for each operation, as well as error handling for various scenarios.

Question 1 (10 points) You are assigned to do the functional (black-box) test for function Report on time:



Function Description:

- Actor: Admin, Agent, Customer
- Pre-condition: The user must have a session, which is responsible for checking out incident ticket that satisfy a set of given conditions.
- Business rule:

The "Request Title" is a required string with a length ranging from 50 to 255 characters. The "Description" is a required string and not exceeding 1000 characters.

The "Attachment" is optional, but it only accepts one of the file types including: jpg, jpeg, png, gif, docx, doc, pdf. The size must not exceed 10MB.

Function Details:

- Normal case
- Abnormal case

1. User enters all information into the fields of Report form and clicks "Create" button.

The system will validate the information entered by the user.

2. Incident ticket is created successfully.

3. Data input is invalid, the system rejects a user asking the user to re-enter the information.

1. Analyze test conditions using the Equivalence Partitioning and Boundary Value

Analysis test design techniques and GT in the table 3 (Question 1 & 2 complex).

2. Identify test cases for each condition and create the test cases (Test Case Design and GT in the table 3.2 (Question 3 & 2 simple)).

3. Implement the test cases (Test Implementation and GT in the table 3 with detailed Pre-condition and Test Case Procedure using table 3.1 (Question 2 & Complex)).

* Notes: Please feel free to include any assumption needed for your answers to be clearer and more accurate.

SWT301 - FALL23 - The final PE

Duration 90 minutes

Instructions:

- You need to know at least one programming language to perform reasonably well on common operations.
- Students are allowed to use IDEs such as NetBeans, IntelliJ IDEA, or similar software to write their code.

Question 1 (10 points) You are assigned to do the following class and test for basic 8 issues in the code (i.e., coding practice, except errors, potential logical errors, etc.). (For practice / template)

```
public class Fibonacci {
    public static long calculateFibonacci(int n) {
        if (n <= 0) {
            System.out.println("The input number is invalid.");
            return 0;
        }
        if (n == 1) {
            System.out.println("The result is 1.");
            return 1;
        }
        long result = 0;
        long first = 0;
        long second = 1;
        for (int i = 2; i <= n; i++) {
            result = first + second;
            first = second;
            second = result;
        }
        System.out.println("The result is " + result);
        return result;
    }
}
```

Question 1 (10 points) Assuming you are assigned to conduct the component test for the method below, please design and create the minimum requirement test cases (Unit Test) and modify the code to achieve 100% statement coverage and 100% decision coverage. (The question 2 complex).

This code defines a *calculator* provides method for value has modulus and an operator *divide* as input. It performs the specified operation (addition, subtraction, multiplication, or division) and returns the result. It includes logging for each operation, as well as error handling for various scenarios.

Question 1 (10 points) You are assigned to do the functional (black-box) test for function Report on time:



Function Description:

- Actor: Admin, Agent, Customer
- Pre-condition: The user must have a session, which is responsible for checking out incident ticket that satisfy a set of given conditions.

- Business rule:

The "Request Title" is a required string with a length ranging from 50 to 255 characters. The "Description" is a required string and not exceeding 1000 characters.

The "Attachment" is optional, but it only accepts one of the file types including:

jpg, jpeg, png, gif, docx, doc, pdf. The size must not exceed 10MB.

Function Details:

- Normal case
- Abnormal case

1. User enters all information into the fields of Report form and clicks "Create" button.

The system will validate the information entered by the user.

2. Incident ticket is created successfully.

3. Data input is invalid, the system rejects a user asking the user to re-enter the information.

1. Analyze test conditions using the Equivalence Partitioning and Boundary Value

Analysis test design techniques and GT in the table 3 (Question 1 & 2 complex).

2. Identify test cases for each condition and create the test cases (Test Case Design and GT in the table 3.2 (Question 3 & 2 simple)).

3. Implement the test cases (Test Implementation and GT in the table 3 with detailed Pre-condition and Test Case Procedure using table 3.1 (Question 2 & Complex)).

* Notes: Please feel free to include any assumption needed for your answers to be clearer and more accurate.