

Lab Practice Assignment 9

Save each test script and screenshots of test result as per question number. Make a zip folder of all the files. Upload the zip file to Moodle.

Problem 1. (20%) Quadric Equation

Quadric_Equation (20 Points)

Consider the Quadric_Equation.java file:

- It determines the nature of roots of a quadratic equation.
- Its input are positive integers (say a,b,c) and values may be from interval [0,200]. The program output may have one of the following phrases: Not a quadratic equation; Real roots; Imaginary roots; Equal roots.

Test the calculateroot() using Equivalence Class Partition

Problem 2. (20%) Commissioning

A rifle salesperson in the former Arizona Territory sold rifle locks, stocks, and barrels made by a gunsmith in Missouri. Locks cost \$45, stocks cost \$30, and barrels cost \$25. The salesperson had to sell at least one complete rifle per month, and production limits were such that the most the salesperson could sell in a month was 70 locks, 80 stocks, and 90 barrels. After each town visit, the salesperson sent a telegram to the Missouri gunsmith with the number of locks, stocks, and barrels sold in that town. At the end of a month, the salesperson sent a very short telegram showing total number of locks, stocks and barrels sold. The gunsmith then knew the sales for the month were complete and computed the salesperson's commission as follows: 10% on sales up to (and including) \$1000, 15% on the next \$800, and 20% on any sales in excess of \$1800. The commission program produced a monthly sales report that gave the total number of locks, stocks, and barrels sold, the salesperson's total dollar sales, and, finally, the commission.

- a) Test the "CommissioningProblem.java" source code using "Boundary Value Analysis" technique.
- b) There is at least one bug in the implementation. Design at least one test case to identify bugs and write a description of the bugs using @DisplayName annotations.

Problem 3. (20%) Client Product

A commercial house needs an application to request products towards customers based on three features: Gender, Residence Type, and Age Group. The Age Groups are - A1 (under 35), A2 (between 35 and 65), A3 (over 65). The commercial house has four products to assign towards clients. Those are W, X, Y and Z. Product W will request to young females. Product X will appeal to all but not to older females. Product Y will request to female city residents. Product Z will request to Male middle-aged shoppers who do not live in cities.

- a) Test this program “Client_Product.java” using the Decision Table based testing. Additionally, submit your 'Decision Table' along with your regular answer
- b) There is at least one bug in the implementation. Design at least one test case to identify bugs and write a description of the bugs using @DisplayName annotations.

Problem 4. (20%) Next Date

NextDate is a function with three variables: month, day, year. It returns the date of the day after the input date. Limitation: 1800-2025

Treatment Summary: If it is not the last day of the month, the next date function will simply, increment the day value. At the end of a month, the next day is 1 and the month is incremented. At the end of the year, both the day and the month are reset to 1, and the year incremented. Finally, the problem of leap year makes determining the last day of a month interesting.

- a. Execute test cases that give you 100% branch coverage (Condition Coverage) for this source code. For this problem, you must also submit your “Decision Table” together with the regular ana.
- b. There is at least one bug in the implementation. Design at least one test case to identify bugs and write a description of the bugs using @DisplayName annotations.

Problem 5. (20%) CaesarShift Cipher

The CaesarShiftCipher class is responsible for applying the encryption method Caesar cipher.

The rule of this method is to shift letters of a message by a given offset.

Let us say we want to shift the alphabet by 3, then letter a would be transformed to letter d, b to e, c to f, and so on.

Here is the complete matching between original and transformed letters for an offset of 3:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

As we can see, once the transformation goes beyond the letter Z, we go back to the start of the alphabet, so that X, Y and Z are transformed into A, B and C, respectively.

public String CaesarShift(String message, int shift) - this method takes input the “message” which have to be encrypted and “shift” which represents the offset. This method will return the encrypted message (The program only takes uppercase alphabets as input).

For example:

Input: message: ATTACKATONCE, Shift: 4

Output: EXXEGOEXSRGI

- a. Execute test cases that give you 100% branch coverage (Condition Coverage) for this source code.
- b. There is at least one bug in the implementation. Design at least one test case to identify bugs and write a description of the bugs using @DisplayName annotations.

~~The end~~