Create a StringBuilder to keep track of negative numbers found during processing.

Iterate through the numbersArray obtained by splitting the input string.

Check if each number is negative (num < 0).

If a negative number is found, append it to the negativeNumbers string.

After the loop, if any negative numbers were found, throw an exception with the message "Negatives not allowed: {negativeNumbers}".

Step 3: Ignore numbers larger than 1000

Modify the existing loop to include a condition that skips numbers larger than 1000 (num > 1000).

This ensures that numbers greater than 1000 are not added to the result.

Step 4: Handle decimal numbers

Add a try-catch block inside the loop to catch NumberFormatException when parsing the numbers.

If a NumberFormatException is caught, throw an exception with the message "Only integers are allowed."

This ensures that decimal numbers or non-numeric characters are not considered valid input.

Step 5: Return the updated returnValue

Add a return statement at the end of the add method to return the updated returnValue.

Write a test class that focuses solely on testing the new requirements (Extended Functionalities):

Step 1: Create a new test class

Create a new test class, such as StringCalculatorTest, specifically for testing the extended functionalities.

Step 2: Write test cases for handling negative numbers

Design test cases to verify that the add method throws an exception when negative numbers are present in the input string.

Ensure that the exception message matches the expected format.

Step 3: Write test cases for ignoring numbers larger than 1000.

Design test cases to verify that the add method ignores numbers larger than 1000 and returns the correct sum.

Include cases with multiple numbers exceeding 1000 and combinations of numbers below and above the threshold.

Step 4: Write test cases for handling decimal numbers

Design test cases to verify that the add method throws an exception when decimal numbers or non-numeric characters are present in the input string.

Ensure that the exception message matches the expected format.

Step 5: Execute the test cases