Tanay Agarwal
CS 401
Test Cases Screenshots
Homework 3

The test cases were automated using Junit. The test report was printed automatically in the test class.

Test cases and result for combine:

```
Test case 1: Combine two words
Inputs: input1 = "Hello", input2 = "World"
Expected Output: "HelloWorld"
Actual Output: "HelloWorld"
Result: Test PASSED
--------------------
Test case 2: Combine two empty strings
Inputs: input1 = "", input2 = ""
Expected Output: ""
Actual Output: ""
Result: Test PASSED
--------------------
Test case 3: Combine word with empty string
Inputs: input1 = "Test", input2 = ""
Expected Output: "Test"
Actual Output: "Test"
Result: Test PASSED
--------------------
Test case 4: Combine empty string with word
Inputs: input1 = "", input2 = "Case"
Expected Output: "Case"
Actual Output: "Case"
Result: Test PASSED
--------------------
Test case 5: Combine two numbers as strings
Inputs: input1 = "123", input2 = "456"
Expected Output: "123456"
Actual Output: "123456"
Result: Test PASSED
--------------------
Test case 6: Combine long strings
Inputs: input1 = "aaaaaaaaaaaaaaaaaaaa", input2 = "bbbbbbbbbbbbbbbbbbbb"
Expected Output: "aaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbb"
Actual Output: "aaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbb"
Result: Test PASSED
--------------------
Test case 7: Combine strings with special characters
Inputs: input1 = "!@#", input2 = "$%^"
Expected Output: "!@#$%^"
Actual Output: "!@#$%^"
Result: Test PASSED
--------------------
Test case 8: Combine strings with spaces
Inputs: input1 = "Hello ", input2 = " World"
Expected Output: "Hello  World"
Actual Output: "Hello  World"
Result: Test PASSED
--------------------
```

Test cases and result for reverse:

```
Test case 9: Reverse a word
Inputs: input1 = "Hello", input2 = "null"
Expected Output: "olleH"
Actual Output: "olleH"
Result: Test PASSED
--------------------
Test case 10: Reverse an empty string
Inputs: input1 = "", input2 = "null"
Expected Output: ""
Actual Output: ""
Result: Test PASSED
--------------------
Test case 11: Reverse a single character
Inputs: input1 = "a", input2 = "null"
Expected Output: "a"
Actual Output: "a"
Result: Test PASSED
--------------------
Test case 12: Reverse a string of numbers
Inputs: input1 = "12345", input2 = "null"
Expected Output: "54321"
Actual Output: "54321"
Result: Test PASSED
--------------------
Test case 13: Reverse a phrase with space
Inputs: input1 = "Test Case", input2 = "null"
Expected Output: "esaC tseT"
Actual Output: "esaC tseT"
Result: Test PASSED
--------------------
Test case 14: Reverse a long string
Inputs: input1 = "aaaaaaaaaaaaaaaaaaaa", input2 = "null"
Expected Output: "aaaaaaaaaaaaaaaaaaaa"
Actual Output: "aaaaaaaaaaaaaaaaaaaa"
Result: Test PASSED
--------------------
Test case 15: Reverse a string with special characters
Inputs: input1 = "!@# $%^", input2 = "null"
Expected Output: "^%$ #@!"
Actual Output: "^%$ #@!"
Result: Test PASSED
--------------------
Test case 16: Reverse a palindrome
Inputs: input1 = "madam", input2 = "null"
Expected Output: "madam"
Actual Output: "madam"
Result: Test PASSED
```

Test cases and result for upperCase:

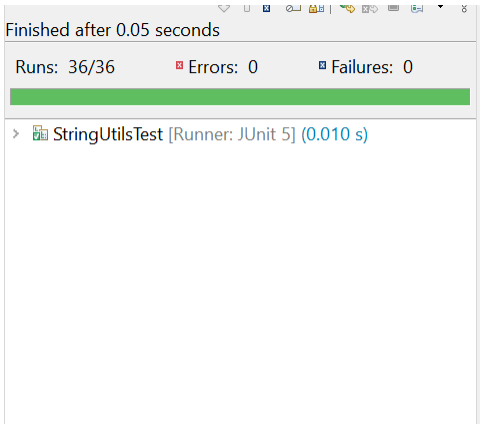
```
Test case 17: Convert lowercase to uppercase
Inputs: input1 = "hello", input2 = "null"
Expected Output: "HELLO"
Actual Output: "HELLO"
Result: Test PASSED
--------------------
Test case 18: Convert empty string to uppercase
Inputs: input1 = "", input2 = "null"
Expected Output: ""
Actual Output: ""
Result: Test PASSED
--------------------
Test case 19: Convert already uppercase string
Inputs: input1 = "ALREADY UPPER", input2 = "null"
Expected Output: "ALREADY UPPER"
Actual Output: "ALREADY UPPER"
Result: Test PASSED
--------------------
Test case 20: Convert mixed case to uppercase
Inputs: input1 = "MixEd CaSe", input2 = "null"
Expected Output: "MIXED CASE"
Actual Output: "MIXED CASE"
Result: Test PASSED
--------------------
Test case 21: Convert alphanumeric to uppercase
Inputs: input1 = "123abc", input2 = "null"
Expected Output: "123ABC"
Actual Output: "123ABC"
Result: Test PASSED
--------------------
Test case 22: Convert string with special characters to uppercase
Inputs: input1 = "!@# abc", input2 = "null"
Expected Output: "!@# ABC"
Actual Output: "!@# ABC"
Result: Test PASSED
--------------------
Test case 23: Convert long lowercase string to uppercase
Inputs: input1 = "aaaaaaaaaaaaaaaaaaaa", input2 = "null"
Expected Output: "AAAAAAAAAAAAAAAAAAAA"
Actual Output: "AAAAAAAAAAAAAAAAAAAA"
Result: Test PASSED
--------------------
Test case 24: Convert mixed case with spaces to uppercase
Inputs: input1 = "Mix Ed CaSe", input2 = "null"
Expected Output: "MIX ED CASE"
Actual Output: "MIX ED CASE"
Result: Test PASSED
```

Test cases and result for trimWhitespace:

```
Test case 25: Trim whitespace from both ends
Inputs: input1 = "  Hello World!  ", input2 = "null"
Expected Output: "Hello World!"
Actual Output: "Hello World!"
Result: Test PASSED
--------------------
Test case 26: Trim whitespace from both ends of a word
Inputs: input1 = " Spaces ", input2 = "null"
Expected Output: "Spaces"
Actual Output: "Spaces"
Result: Test PASSED
--------------------
Test case 27: Trim string with no whitespace
Inputs: input1 = "NoSpaces", input2 = "null"
Expected Output: "NoSpaces"
Actual Output: "NoSpaces"
Result: Test PASSED
--------------------
Test case 28: Trim string with only whitespaces
Inputs: input1 = "    ", input2 = "null"
Expected Output: ""
Actual Output: ""
Result: Test PASSED
--------------------
Test case 29: Trim string with multiple internal spaces
Inputs: input1 = "  Multiple   Spaces  ", input2 = "null"
Expected Output: "Multiple   Spaces"
Actual Output: "Multiple   Spaces"
Result: Test PASSED
--------------------
Test case 30: Trim long string with whitespaces
Inputs: input1 = "                    Text                    ", input2 = "null"
Expected Output: "Text"
Actual Output: "Text"
Result: Test PASSED
--------------------
Test case 31: Trim string with newline characters
Inputs: input1 = "
         Hello
", input2 = "null"
Expected Output: "Hello"
Actual Output: "Hello"
Result: Test PASSED
--------------------
Test case 32: Trim string with mixed whitespace characters
Inputs: input1 = "        Mixed
  ", input2 = "null"
Expected Output: "Mixed"
Actual Output: "Mixed"
Result: Test PASSED
```

Test Summary and multi method tests:

```
Test case 33: Combine and reverse
Inputs: input1 = "Hello", input2 = "World"
Expected Output: "dlroWolleH"
Actual Output: "dlroWolleH"
Result: Test PASSED
--------------------
Test case 34: Combine, reverse, and uppercase
Inputs: input1 = "Hello", input2 = "World"
Expected Output: "DLROWOLLEH"
Actual Output: "DLROWOLLEH"
Result: Test PASSED
--------------------
Test case 35: Combine, trim, and uppercase
Inputs: input1 = "  Hello", input2 = "World  "
Expected Output: "HELLOWORLD"
Actual Output: "HELLOWORLD"
Result: Test PASSED
--------------------
Test case 36: Combine, reverse, and trim
Inputs: input1 = " Hello", input2 = "World "
Expected Output: "dlroWolleH"
Actual Output: "dlroWolleH"
Result: Test PASSED
--------------------
Total test cases run: 36
Number of cases passed: 36
Number of cases failed: 0
```




```
81
82      @BeforeClass
83      public static void setUp(
84          System.out.println("
85          System.out.println("
86          totalTestCases = 0;
87          failedCaseNumber = 0
88          passedCaseNumber = 0
89      }
90
91      @Before
92      public void setUp() {
93          totalTestCases++;
```

Issues encountered during testing:
I encountered three types of issues during testing.

1) The test results that were printed initially did not have any details. I had to use @BeforeClass, @Before, @After, @AfterClass and print test results on the console including final test statistics before I could figure out what is going on easily.

2) I found one logical error in my implementation of the trimWhitespace method. I ended up removing whitespaces in the middle of the string and not just trimming the beginning and end of the string.

3) I then changed my code in the class being tested to custom code for other methods to try and introduce errors. However the new implementations worked just fine.

4) Most of my errors were due to the fact that I did not correctly enter the expected result. The class was working properly but the test showed an error because of this.

Test coverage:
I ran 36 test cases. Eight test cases were for each method and four test cases used a combination of methods. The test cases were designed to cover normal use of the method as well as uses that one would not encounter during normal operation. The test coverage could have been expanded especially the combination testing but I thought this was adequate testing.

I did not use any tool that estimates whether all the different combinations in the code are exercised during the test. This is one area of improvement that could have been implemented.

Overall quality assessment of the string operations class:
All the 36 test cases passed. This shows the overall quality of string operations class is good. I could have implemented a tool to assess test coverage. This would have provided a higher confidence in the quality of the string operations class.