
System Requirements Specification Index

For

String Manipulation with Special Characters

Version 1.0

TABLE OF CONTENTS

1	Project Abstract	3
2	Common Constraints	3
3	Template Code Structure	4
3.1	Package: com.yaksha.assignment.StringWithSpecialCharactersAssignment	4
4	Execution Steps to Follow	5

USE CASE DESCRIPTION

System Requirements Specification

1 PROJECT ABSTRACT

This project focuses on manipulating strings that contain special characters. Special characters are non-alphanumeric symbols like punctuation marks, spaces, escape sequences, or any character other than letters and numbers.

2 ASSESSMENT TASKS

Task 1: Declare a String Containing Special Characters:

→ Declare a string that includes special characters like punctuation marks, spaces, and escape sequences. A typical example could be "Hello, World! How's everything?", which contains spaces, punctuation, and a quotation mark.

→ Expected Outcome:

The string should contain non-alphanumeric characters that will be manipulated later.

Task 2: Replace Special Characters in a String:

→ Replace non-alphanumeric characters (such as punctuation marks) with spaces or remove them entirely from the string. This can be done using regular expressions in Java, allowing you to either remove or substitute special characters.

→ Expected Outcome:

The string should be transformed to remove or replace unwanted special characters. For example, the string "Hello, World! How's everything?" could become "Hello World Hows everything" after removing punctuation marks.

Task 3: Escape Special Characters in a String:

→ Escape special characters such as double quotes or backslashes in a string. Escape sequences are used to ensure that these characters are interpreted as part of the string rather than as special symbols in the code.

→ Expected Outcome:

The string should be transformed to properly escape any special characters. For example, if the string contains a quotation mark (") or a backslash (\), they should be escaped as \" and \\, respectively.

Task 4: Trim Whitespace and Special Characters:

→ Remove any leading or trailing whitespace or special characters from the string. This operation is important for sanitizing input strings, especially in user input or data processing scenarios.

→ Expected Outcome:

The string should have no leading or trailing spaces or special characters, providing a clean version of the string. For example, " Hello, World! " should become "Hello, World!" after trimming.

Task 5: Check if a String Contains Special Characters:

→ Check if the string contains any special characters, such as punctuation, whitespace, or other non-alphanumeric symbols. This can be done using regular expressions to look for patterns in the string.

→ Expected Outcome:

The result should be a boolean indicating whether the string contains special characters. For instance, checking the string "Hello, World!" should return true because it contains a comma and an exclamation mark.

3 TEMPLATE CODE STRUCTURE

3.1 PACKAGE: COM.YAKSHA.ASSIGNMENT.STRINGWITHSPECIALCHARACTERSASSIGNMENT

Resources

Class/Interface	Description	Status
StringWithSpecialCharactersAssignment (class)	<ul style="list-style-type: none">Main class demonstrating string operations involving special characters, such as replacing, escaping, trimming, and checking for special characters in a string.	Need to be implemented.

4 EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) □ Terminal □New Terminal.
3. This editor Auto Saves the code.
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To run your project use command:
mvn compile exec:java
-Dexec.mainClass="com.yaksha.assignment.StringWithSpecialCharactersAssignment"
7. To test your project test cases, use the command
mvn test
8. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.