System Requirements Specification Index

For

String Comparison and Conversion

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1 Project Abstract

This project will assess knowledge of string comparison and conversion methods in Java. You need to implement string comparison and conversion operations using built-in Java methods.

2 Assessment Tasks

1. Declare two string variables that will be used for comparison and conversion operations.

2. Use string comparison methods:
   equals(), equalsIgnoreCase(), compareTo().
   Examples:
   - Check if two strings are equal, considering case sensitivity and ignoring case sensitivity.
   - Compare two strings lexicographically using compareTo().

3. Apply conversion methods:
   valueOf(), trim(), split(), toCharArray().
   Examples:
   - Declare a string variable.
   - Convert other data types to strings using valueOf().
   - Trim leading and trailing spaces from a string using trim().
   - Split a string into an array of substrings using split().
   - Convert a string into an array of characters using toCharArray().

3 Template Code Structure

3.1 Package: com.yaksha.assignment.StringComparisonConversionAssignment

Resources

Class/Interface	Description	Status
StringComparisonConversionAssignment (class)	● Main class demonstrating string comparison operations such as: `equals`, `equalsIgnoreCase`, `compareTo.` ● And string conversion operations like: `valueOf`, `trim`, `split`, and `toCharArray`.	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.StringComparisonConversionAssignment"
   ```

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.**

9. **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.**