System Requirements Specification Index

For

String Indexing and Concatenation

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 Project Abstract

You will be required to demonstrate the use of string indexing techniques and various string concatenation methods like:

- String Indexing: Accessing characters from specific positions and finding positions of substrings.
- String Concatenation: Combining multiple strings using different techniques.

2 Assessment Tasks

1. String Indexing Tasks:
    - Declare a string variable with value, i.e. "Hello, Java World!"

    - Accessing Characters in a String:
        a. Using charAt(index) to access the character at a specified index, i.e. 7.

    - Finding Index Positions:
        a. Using indexOf(substring) to find the index of the first occurrence of a substring, i.e. Java.
        b. Using lastIndexOf(substring) to find the index of the last occurrence of a substring, i.e l.

    - Extracting Substrings:
        a. Using substring(startIndex, endIndex) to extract a substring between the specified start and end indices, i.e. 7 and 11 respectively.

2. String Concatenation Tasks:
    - Declare two string variables with values “Hello” and “Java”.

    - Concatenating Strings Using concat():
        a. Using the concat() method to combine strings, i.e. ” “ & any one string variable.

    - Concatenating Strings Using + Operator:
        a. Using the + operator to concatenate strings in Java, i.e. add space between two string variables.

- Efficient Concatenation Using StringBuilder:
    a. Using the StringBuilder class to append strings efficiently, i.e. add space between two string variables.

3 Template Code Structure

3.1 Package: com.yaksha.assignment.StringIndexingConcatenationAssignment

Resources

Class/Interface	Description	Status
StringIndexingConcatenationAssignment (class)	● Main class demonstrating string indexing operations like: `charAt`, `indexOf`, `lastIndexOf`, `substring`. ● And string concatenation operations using: `concat`, `+` operator, and `StringBuilder`.	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.StringIndexingConcatenationAssignment"
   ```

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.**