System Requirements Specification Index

For

Comparison Operators

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 Project Abstract

Create a Java program that uses comparison operators (==, !=, <, >, <=, >=) to compare two variables and print the results.

2 Assessment Tasks

1. **Declare 2 variables:**
   - A variable named `a` of `int` datatype, initialized with the value `10`.
   - A variable named `b` of `int` datatype, initialized with the value `20`.

2. **Perform Comparison Operations:**
   Perform the following comparison operations using the declared variables `a` and `b`:

   - **Equality Check (`==`)**:
     1) Check if `a` is equal to `b` using `a == b`.
     2) Store the result in a boolean variable named `isEqual`.
   - **Inequality Check (`!=`)**:
     1) Check if `a` is not equal to `b` using `a != b`.
     2) Store the result in a boolean variable named `isNotEqual`.
   - **Greater Than Check (`>`)**:
     1) Check if `a` is greater than `b` using `a > b`.
     2) Store the result in a boolean variable named `isGreaterThan`.
   - **Less Than Check (`<`)**:
     1) Check if `a` is less than `b` using `a < b`.
     2) Store the result in a boolean variable named `isLessThan`.
   - **Greater Than or Equal Check (`>=`)**:
     1) Check if `a` is greater than or equal to `b` using `a >= b`.
     2) Store the result in a boolean variable named `isGreaterThanOrEqual`.
   - **Less Than or Equal Check (`<=`)**:
     1) Check if `a` is less than or equal to `b` using `a <= b`.
     2) Store the result in a boolean variable named `isLessThanOrEqual`.

   Print the Results:
   - Print the results of the comparison operations i.e, `isEqual`, `isNotEqual`, `isGreaterThan`, `isLessThan`, `isGreaterThanOrEqual`and `isLessThanOrEqual` with appropriate labels in separate lines as shown in the expected output.

Expected Output:

```
Is a equal to b? false

Is a not equal to b? true

Is a greater than b? false

Is a less than b? true

Is a greater than or equal to b? false

Is a less than or equal to b? true
```

3 Template Code Structure

3.1 Package: com.yaksha.assignment.ComparisonOperatorsAssignment

Resources

Class/Interface	Description	Status
ComparisonOperatorsAssignment (class)	● Main class containing the logic to perform and demonstrate comparison operators (==, !=, >, <, >=, <=).	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.ComparisonOperatorsAssignment"
   ```
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.**