System Requirements Specification Index

For

Basic Logical Operators

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1 Project Abstract

In this project, you will demonstrate your understanding of Java Logical Operators. The task will focus on three essential logical operators used in Java:

- AND (&&): Evaluates to true if both conditions are true.
- OR (||): Evaluates to true if at least one of the conditions is true.
- NOT (!): Negates or inverts the boolean value of a condition.

These logical operators are fundamental for controlling the flow of execution in programs, particularly when you need to combine conditions or make decisions based on multiple factors.

2 Assessment Tasks

Task 1:

1. **Declare 2 variables:**
   - A variable named `a` of `boolean` datatype, initialized with the value `true`.
   - A variable named `b` of `boolean` datatype, initialized with the value `false`.

2. **Perform Logical Operations:**
   Use the boolean variables `a` and `b` to perform the following logical operations:

   - **AND Operation (`&&`)**:
     1) Compute the logical AND of `a` and `b` (`a && b`) and store the result in `andResult` of `boolean` datatype.
   - **OR Operation (`||`)**:
     1) Compute the logical OR of `a` and `b` (`a || b`) and store the result in `orResult` of `boolean` datatype.
   - **NOT Operation (`!`)**:
     1) Compute the logical NOT of `a` (`!a`) and store the result in `notResultA` of `boolean` datatype.
     2) Compute the logical NOT of `b` (`!b`) and store the result in `notResultB` of `boolean` datatype.

   Print the Results:
   - Print the results of each logical operation i.e, `andResult`, `orResult`, `notResultA` and `notResultB` with appropriate labels in separate lines as shown in the expected output.

Task 2:

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

4. **Perform Combined Logical Operations:**
   Use the integer variables `x` and `y` to perform the following combined logical operations:

    - **Complex Condition**:
        1) Evaluate the condition `((x > 5 && x < 15) || (y > 10 && y < 25))`.
        2) Store the result in a `boolean` variable named `complexCondition`.
    - **Negation Condition**:
        1) Evaluate the condition `!(x == 10)`.
        2) Store the result in a boolean variable named `negationCondition`.
5. **Print the Results:**
    - Print the results of each combined logical operation i.e, `complexCondition`, and `negationCondition` with appropriate labels in separate lines as shown in the expected output.

Expected Output:

```
AND (a && b): false

OR (a || b): true

NOT (a): false

NOT (b): true

Complex Condition ((x > 5 && x < 15) || (y > 10 && y < 25)): true

Negation Condition (!(x == 10)): false
```

3 Template Code Structure

3.1 Package: com.yaksha.assignment.LogicalOperatorsAssignment

Resources

Class/Interface	Description	Status
LogicalOperatorsAssignment (class)	• Main class containing the logic to demonstrate logical operators: AND (&&), OR (||), NOT (!).	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. **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.**
5. **To run your project use command:**

   ```
   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.LogicalOperatorsAssignment"
   ```

   ***If it asks for the password, provide password : pass@word1**

6. **To test your project test cases, use the command**

   ```
   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn test
   ```

   ***If it asks for the password, provide password : pass@word1**