System Requirements Specification Index

For

Advanced Logical Operators

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION
System Requirements Specification

1 Project Abstract

In this project, you need to demonstrate your understanding of Java Bitwise Operators by performing various operations directly on the binary representations of integers. Your task is to implement and use the following bitwise operators in your program:

- AND (&)
- OR (|)
- XOR (^)
- NOT (~)
- Left Shift (<<)
- Right Shift (>>)

Your implementation should include examples of each operator, showcasing their functionality and the results of their application on given input values.

2 Assessment Tasks

Task 1: Declare Two Integer Variables:

Declare two integer variables, a and b. These variables will be used to perform bitwise operations.

Task 2: Perform Bitwise Operations:

Apply the following bitwise operators to the variables a and b:

- AND (&): Perform a bitwise AND operation between a and b.
- OR (|): Perform a bitwise OR operation between a and b.
- XOR (^): Perform a bitwise XOR operation between a and b.
- NOT (~): Perform a bitwise NOT operation on a.
- Left Shift (<<): Perform a left shift on a by 2 positions.
- Right Shift (>>): Perform a right shift on a by 1 position.
- Print the Results: Output the result of each operation.

Task 3: Use Bitwise Operators in a Combination:

Combine bitwise operators to create more complex expressions:

- Shift a number and then apply AND or OR to the result.
- Negate a number and apply other bitwise operations.

Example:

- Declare 2 integers x and y.
- Shift x to the left by 1 position and then apply an AND operation with y.
- Negate x and print the result.

3 Template Code Structure

3.1 Package: com.yaksha.assignment.TestBitwiseOperatorsAssignment

Resources

Class/Interface	Description	Status
TestBitwiseOperatorsAssignment (class)	• Main class demonstrating the use of bitwise operators: AND (&), OR (|), XOR (^), NOT (~), Left Shift (<<), and Right Shift (>>).	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.TestBitwiseOperatorsAssignment"
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.**