System Requirements Specification Index

For

Advanced Logical Operators

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

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:

1. **Declare 2 variables:**
   - A variable named `a` of `int` datatype, initialized with the value `5`.
   - A variable named `b` of `int` datatype, initialized with the value `3`.

2. **Perform Bitwise Operations:**
   Use the integer variables `a` and `b` to perform the following bitwise operations:

   - **Bitwise AND (&)**:
     1) Compute the bitwise AND of `a` and `b` (`a & b`) and store the result in `andResult` of `int` datatype.
   - **Bitwise OR (|)**:
     1) Compute the bitwise OR of `a` and `b` (`a | b`) and store the result in `orResult` of `int` datatype.
   - **Bitwise XOR (^)**:
     1) Compute the bitwise XOR of `a` and `b` (`a ^ b`) and store the result in `xorResult` of `int` datatype.
   - **Bitwise NOT (~)**:
     1) Compute the bitwise NOT of `a` (`~a`) and store the result in `notResultA` of

int datatype.

- **Left Shift (<<)**:
  1) Shift the bits of a to the left by 2 (a << 2) and store the result in leftShiftResult of int datatype.
- **Right Shift (>>)**:
  1) Shift the bits of a to the right by 1 (a >> 1) and store the result in rightShiftResult of int datatype.

Print the Results:

- Print the results of each bitwise operation i.e, andResult, orResult, xorResult, notResultA, leftShiftResult and rightShiftResult 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 8.
   - A variable named b of int datatype, initialized with the value 4.

4. **Perform Combined Bitwise Operations:**
   Use the variables x and y to perform the following combined bitwise operations:

   - **Complex Expression**:
     1) Shift x to the left by 1 (x << 1) and then perform bitwise AND with y (& y).
     2) Store the result in a variable named complexResult of int datatype.
   - **Negation**:
     1) Compute the bitwise NOT of x (~x).
     2) Store the result in a variable named negationResult of int datatype.

Print the Results:

- Print the results of each combined bitwise operation i.e, complexResult, and negationResult with appropriate labels in separate lines as shown in the expected output.

Expected Output:

```
AND (a & b): 1
OR (a | b): 7
XOR (a ^ b): 6
NOT (~a): -6
Left Shift (a << 2): 20
Right Shift (a >> 1): 2
Complex Result ((x << 1) & y): 0
Negation Result (~x): -9
```

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.**