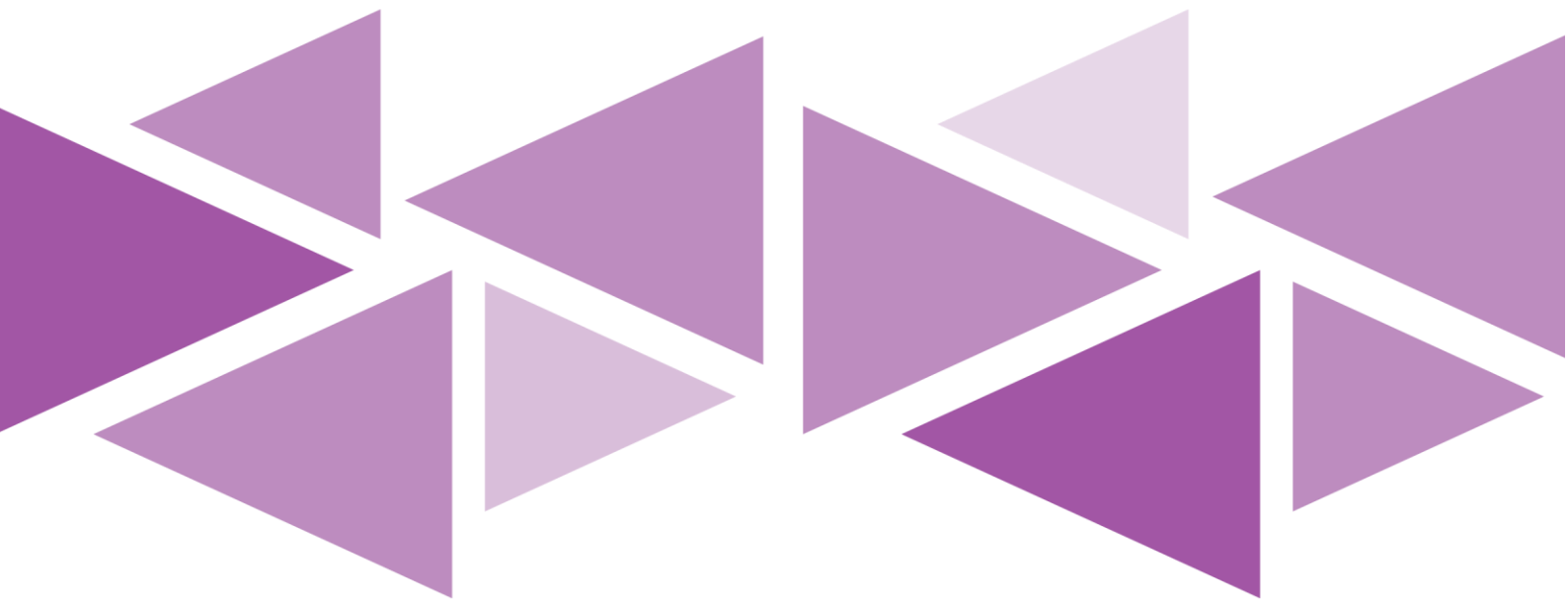


BITWISE & SHIFT OPERATOR ASSIGNMENT



**Submitted by
Harish Kumbhar**

Java Bitwise and Shift Operators

In Java, bitwise operators perform operations on integer data at the individual bit-level. The integer data includes **byte, short, int, long**.

There are 7 operators to perform bit-level operations in Java.

Operator	Description
	Bitwise OR
&	Bitwise AND
^	Bitwise XOR
~	Bitwise Complement
<<	Left Shift
>>	Signed Right Shift
>>>	Unsigned Right Shift

1. Java Bitwise OR Operator

The bitwise **OR [|]** operator returns 1 if at least one of the operands is 1, Otherwise, it returns 0.

OR (A | B) Truth Table

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Example

6 = 0110 (In Binary)

5 = 0101 (In Binary)

Bitwise OR Operation of 6 and 5

0110

0101

0111 = 7 (In Decimal)

2. Java Bitwise AND Operator

The bitwise AND [&] operator returns 1 if and only if both the operands are 1, Otherwise, it returns 0.

OR (A & B) Truth Table

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Example

6 = 0110 (In Binary)

5 = 0101 (In Binary)

Bitwise AND Operation of 6 and 5

0110

0101

0100 = 4 (In Decimal)

3. Java Bitwise Inclusive OR / XOR Operator

The bitwise XOR ^ operator returns 1 if and only if one of the operands is 1. However, if both the operands are 0 or if both are 1, then the result is 0.

XOR (A ^ B) Truth Table

A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0

Example

6 = 0110 (In Binary)

5 = 0101 (In Binary)

Bitwise XOR Operation of 6 and 5

0110

0101

0011 = 3 (In Decimal)

4. Java Bitwise Complement Operator

The bitwise complement operator is a unary operator (works with only one operand). It is denoted by \sim .

It changes binary digits **1** to **0** and **0** to **1**.

It is important to note that the bitwise complement of any integer **N** is equal to **-(N + 1)**.

For example, consider an integer **35**. As per the rule, the bitwise complement of **35** should be **-(35 + 1) = -36**. Now let's see if we get the correct answer or not.

In binary arithmetic, we can calculate the binary negative of an integer using 2's complement.

1's complement changes **0** to **1** and **1** to **0**. And, if we add **1** to the result of the 1's complement, we get the 2's complement of the original number. For example,

Example

6 = 0110 (In Binary)

1's Complement

Bitwise Complement Operation of 6

0110

1001 = -7 (In Decimal)

5. Java Left Shift Operator

The left shift operator shifts all bits towards the left by a certain number of specified bits. It is denoted by $<<$.

When we perform a 1 bit left shift operation on it, each individual bit is shifted to the left by **1** bit. As a result, the left-most bit (most-significant) is discarded and the right-most position (least-significant) remains vacant. This vacancy is filled with **0s**.

Example

6 = 0110 (In Binary)

5 = 0101 (In Binary)

Bitwise Left Shift Operation of 6 << 1

0110

1100 = 12 (In Decimal)

6. Java Signed Right Shift Operator

The signed right shift operator shifts all bits towards the right by a certain number of specified bits. It is denoted by >>.

When we shift any number to the right, the least significant bits (rightmost) are discarded and the most significant position (leftmost) is filled with the sign bit.

Example

6 = 0110 (In Binary)

Bitwise Signed Right Shift Operation of 6 >> 1

0110

0011 = 3 (In Decimal)

7. Java Unsigned Right Shift Operator

Java also provides an unsigned right shift. It is denoted by `>>>`.

Here, the vacant leftmost position is filled with **0** instead of the sign bit. For example,

Example

6 = 0110 (In Binary)

Bitwise Unsigned Right Shift Operation of 6 >>> 1

0110

0011 = 3 (In Decimal)

Program

```
public class Bitwise {  
    public static void main(String[] args) {  
        int a = 6, b = 5;  
  
        // There are 7 Bitwise operators  
        System.out.println("Bitwise AND Result Is : "+(a & b));  
        System.out.println("Bitwise OR Result Is : "+(a | b));  
        System.out.println("Bitwise XOR Result Is : "+(a ^ b));  
        System.out.println("Bitwise Complement Result Is : "+(~ a));  
        System.out.println("Bitwise Complement Result Is : "+(~ b));  
        System.out.println("Bitwise Left Shift Result Is : "+(a <<  
            1));  
        System.out.println("Bitwise Signed Right Shift Result Is :  
            "+(a >> 1));  
        System.out.println("Bitwise Unsigned Right Shift Result Is :  
            "+(a >>> 1));  
    }  
}
```

Output:

Bitwise AND Result Is : 4

Bitwise OR Result Is : 7

Bitwise XOR Result Is : 3

Bitwise Complement Result Is : -7

Bitwise Complement Result Is : -6

Bitwise Left Shift Result Is : 12

Bitwise Signed Right Shift Result Is : 3

Bitwise Unsigned Right Shift Result Is : 3

Harish Kumbhar