

# **BITWISE OPERATOR IN JAVA**

**Submitted by**

Name : Yadhukrishna Mk

email : yadhu8824@gmailcom

# SHIFT OPERATOR

The operator which is used to shift the bit patterns right or left is called shift operator in java.

The general form of shift expression is as follows:

left\_operand op n

where,

left\_operand → left operand that can be of integral type.

op → left shift or right shift operator.

n → number of bit positions to be shifted. It must be of type int only.

For example:

10 << 2

## Types of Shift Operator in Java

There are two types of shift operators in Java. They are:

1. Left shift operator
2. Right shift operator

## Left Shift Operator in Java - <<

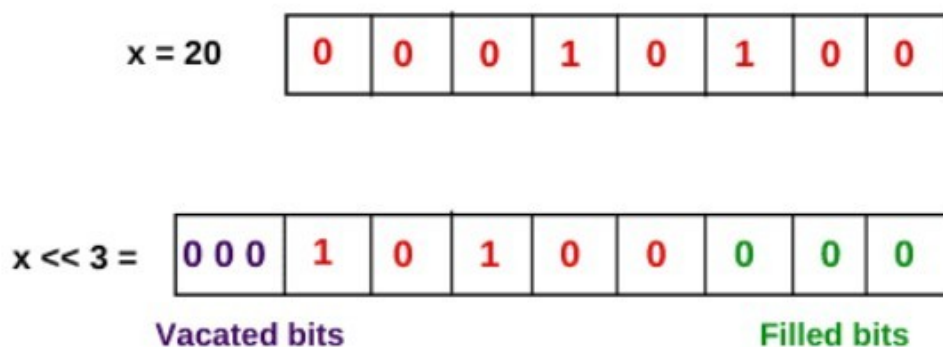
The operator that shifts the bits of number towards left by n number of bit positions is called left shift operator in Java.

This operator is represented by a symbol <<, read as double less than. If we write  $x \ll n$ , it means that the bits of x will be shifted towards left by n positions.

If  $\text{int } x = 20$ . Calculate x value if  $x \ll 3$ .

The value of x is  $20 = 00010100$  (binary format). Now  $x \ll 3$  will shift the bits of x towards left by 3 positions. Due to which leftmost 3 bits will be lost.

Hence, after shifting, bits of x is  $10100000$  that is 160 in decimal form



## **Right Shift Operator in Java - >>**

The operator that shifts the bits of number towards the right by n number of bit positions is called right shift operator in Java.

The right shift operator in java is represented by a symbol >>, read as double greater than.

If we write `x >> n`, it means that the bits of x will be shifted towards right by n positions.

There are two types of right shift operators in java:

1. Signed right shift operator (>>)
2. Unsigned right shift operator (>>>)

## Signed Right Shift Operator

The signed right shift operator  $\gg$  shifts bits of the number towards the right and also reserves the sign bit, which is leftmost bit. A sign bit represents the sign of a number.

If the sign bit is 0 then it represents a positive number. If the sign bit is 1, it represents a negative number.

If the number is positive, the leftmost position is filled with 0. If the number is negative, the leftmost position is filled with 1.

The signed shift operator uses the same sign as used in the number before shifting of bits.

Example.

1. If  $\text{int } x = 10$  then calculate  $x \gg 3$  value.

The value of  $x$  is  $10 = 00001010$ . Since the number is positive, the leftmost bit position will be filled with 0.

Now  $x \gg 3$  will shift the bits of  $x$  towards the right by 3 positions. The rightmost 3 bits will be lost due to shifting. Hence, after shifting, bits of  $x$  is  $00000001$  that is 1 in decimal form.

$x = 10$

0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---

$x \gg 3 =$

0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---

Filled bits

Vacated bits