

# BITWISE OPERATORS IN JAVA

## (RIGHT SHIFT AND LEFT SHIFT)

## Operators

Operators are the mechanism to instruct the processor to perform particular operation. They are special symbols that perform specific operations on one, two, or three operands, and then return a result.

There are several types of operators based on operations performed by the processor. All operators are categorized into 3 categories namely:

1. Unary Operator
2. Binary Operator
3. Ternary Operator

## Bitwise Operators

A bitwise operator is an operator used to perform bitwise operations on bit patterns or binary numerals that involve the manipulation of individual bits.

Bitwise operators are classified into five categories, namely:

- 1) Bitwise AND
- 2) Bitwise exclusive OR
- 3) Bitwise inclusive OR
- 4) Bitwise Compliment
- 5) Bit Shift Operators

## Bit Shift Operators

By shifting the bits of its first operand right or left, a shift operator performs bit manipulation on data.

## Right Shift Operator (>>)

Right shift operator shifts all of the bits in a value to the right a specified number of times. That is, the >> (right shift operator) moves all of the digits in the specified value to the right the number of bit positions specified by the number.

Let us look at an example of right shift operator. Let us shift the value of '32' by 3 and find the result.

The binary value of 32 is 100000. When we shift by 3 the right shift operator shifts the decimal point by digits towards right resulting 100 whose decimal value is 4. Hence the result is 4.

### Java code for right shift operator :

```
class Logical
{
    public static void main(String []args)
    {
        int x=32; // binary value of 32 is 100000
        int y=x>>3;
        System.out.println("The right shifted value is " +y);
    }
}
```

## Left Shift Operator (<<)

Left shift operator shifts all of the bits in a value to the left a specified number of times. That is, the << (left shift operator) moves all of the in the specified value to the left the number of bit positions specified by the number.

Let us look at the same example of left shift operator. Let us shift the value of '32' by 3 and find the result.

The binary value of 32 is 100000. When we shift by 3 the left shift operator shifts the decimal point by digits towards left resulting 100000000 whose decimal value is 256. Hence the result is 256.

### Java code for left shift operator:

```
class Logical
{
    public static void main(String []args)
    {
        int x=32; // binary value of 32 is 100000
        int y=x<<3;
        System.out.println("The left shifted value is " +y);
    }
}
```

