

## ***Bit-Wise Operator In Java***

**Submitted by:**

***Name:*** Santhosh S

***Batch:*** July-A2

***Email:*** [santhoshsanthosh86920@gmail.com](mailto:santhoshsanthosh86920@gmail.com)

## **Shift Operator:**

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

## **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:**

The left shift operator shifts all bits towards the left by a certain number of specified bits.

This operator is represented by a symbol (<<).

### **Example:**

If int x = 4. Calculate x value if x << 1.

The value of x is 4 = 0 1 0 0 (binary format).

Now x <<1 will shift the bits of x towards left by 1 positions. Due to which leftmost 1 bits will be lost.

Hence, after shifting, bits of x is 1 0 0 0 that is 8 in decimal form

### **Code:**

```
public class Eg {  
    public static void main(String[] args)  
{  
        int a=4;  
        System.out.println(a<<1);  
    }  
}
```

## **Output:**

8

## **RIGHT SHIFT OPERATOR:**

The right shift operator shifts all bits towards the right by a certain number of specified bits.

This operator is represented by a symbol ( $\gg$ ).

There are two types of right shift operators:

1. Signed Right Shift ( $\gg$ )
2. Unsigned Right Shift ( $\ggg$ )

### **1. Signed Right Shift Operator ( $\gg$ ):**

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

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.

The signed right shift operator preserves the sign of the number being shifted.

## **Example:**

If  $\text{int } x = 4$  then calculate  $x \gg 1$  value.

The value of  $x$  is  $4 = 0100$ . Since the number is positive, the leftmost bit position will be filled with 0.

Now  $x \gg 1$  will shift the bits of  $x$  towards the right by 1 positions. The rightmost 1 bits will be lost due to shifting.

Hence, after shifting, bits of  $x$  is  $0010$  that is 2 in decimal form.

### Code:

```
public class Eg {  
    public static void main(String[] args)  
{  
        int a=4;  
        System.out.println(a>>1);  
    }  
}
```

### Output:

2

## 2. Unsigned Right Shift Operator (>>>):

The unsigned 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 0's.

### Example:

If int x = 240 then calculate x >>> 2 value. The value of x is 240 = 1 1 1 1 0 0 0 0.

The left operand value is moved right by the number of bits specified by the right operand and shifted bits are filled up with zeros.

Excess bits shifted off to the right are discarded.

Therefore, before shifting the bits the decimal value of a is 240 and after shifting the bits x is 0 0 1 1 1 1 0 0 in decimal form.

**Code:**

```
public class Eg {  
    public static void main(String[] args)  
{  
        int a=8;  
        System.out.println(a>>>1);  
    }  
}
```

**Output:**

**4**