

Bitwise operator

Bitwise operators are used to performing the manipulation of individual bits of a number they can be used with any type like (char, int, float)

They are used when performing update and query operation of the binary tree.

There are some types of bitwise operators

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)
- Bitwise Complement (~)

This are bitwise operators and there are Bit-Shift Operator (Shift Operator)

Bit-Shift Operator (Shift Operator)

Shift operator are used to shift the bits of a number left or right thereby multiplying or dividing the number by two, respectively they can be used when we have to multiply to divide a number by two

There are some types of Bit-Shift Operator

- Signed Right shift operator (>>)
- Unsigned Right shift operator (>>>)
- Signed Left shift operator(<<)
- Unsigned Left shift operator (<<<)

Now we here going to see

- Right shift operator (>>)
- Left shift operator (<<)

Right shift operator

The Right Shift Operator **shift the bits of the number towards right a specified n number of position**. Right Shift Operator represented by the symbol (>>), read as double greater than. When we write **a>>b**, The meaning is to shift the bits **a towards the right b** specified position.

Example: now a=5 *convert the number into binary number*

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|-----|----|----|----|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Now a=5 Right Shift operator has done, a=1

Let see this program:

```
right shift.java

Class right shift
{
    public static void main (String []args)
    {
        Int a=5;
        Int b=10;

        System.out.println("a>>2: "+(a>>2));
        System.out.println("b>>2: "+(b>>2));
        System.out.println("(a>>2) + (b>>2): "+((a>>2)+(b>>2)));
        System.out.println("(a>>2) - (b>>2): "+((a>>2)-(b>>2)));
    }
}
```

Black color – print statement

Red color – main methods

Green color – class methods

Blue color – is saving extension the file

Output

a>>2: 1

b>>2: 2

(a>>2) + (b>>2): 3

(a>>2) - (b>>2): -1

This is Right Shift Operators

Left Shift Operator

The Left-Shift operator causes the bits in shift expression to be shifted to the left by the number of position is specified by additive expression. The bit position that has been vacated by the shift operation are zero – filled. Left Shift operator represented by the symbol (<<).

Example: now a=10 *convert the number into binary number*

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|-----|----|----|----|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

Now a=10 Left Shift operator has done, a=40

Now b=5 *convert the number into binary number*

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|-----|----|----|----|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

Now b=5 Left Shift operator has done, a=20

left shift.java

```
Class left shift
{
    public static void main (String []args)
    {
        Int a=10;
        Int b=5;
        System.out.println("a<<2: "+(a<<2));
        System.out.println("b<<2: "+(b<<2));
        System.out.println("(a<<2) + (b<<2): "+((a<<2)+(b<<2)));
        System.out.println("(a<<2) - (b<<2): "+((a<<2)-(b<<2)));
    }
}
```

Black color – print statement

Red color – main methods

Green color – class methods

Blue color – is saving extension the file

Output

a<<2: 40

b<<2: 20

(a<<2) + (b<<2): 60

(a<<2) - (b<<2): 20

This is Left Shift Operators

Thank you