Bitwise operator:-

Bitwise operators are used to performing the manipulation of individual bits of a number. They can be used with any integral type (char, short, int, etc.).

Bit-Shift Operators (Shift Operators):-

Shift operators 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 or divide a number by two.

Lift-Shift Operator(<<):-

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

Example:-

Let x= 8

Binary number of 8 is 1000

1	0	0	0

if x<<1 -- one 0 is added at the end

1	0	0	0	0

Now the value of x is 16.

if x<<2 -- two 0 is added at the end

1	0	0	0	0	0

Now the value of x is 32.

Right-Shift Operator:-

The left operands value is moved right by the number of bits specified by the right operand.

Java supports two types of right shift operators.

The >> operator is a signed right shift operator.

The >>> operator is an unsigned right shift operator.

Signed Right-Shift Operator(>>):-

The signed right shift operator '>>' uses the sign bit to fill the trailing positions. For example, if the number is positive then 0 will be used to fill the trailing positions and if the number is negative then 1 will be used to fill the trailing positions.

Unsigned Right Shift Operator(>>>):-

The unsigned right shift operator '>>>' do not use the sign bit to fill the trailing positions. It always fills the trailing positions by 0s.

Examples:-

Let x=28

Binary number of 28 is 011100

0	1	1	1	0	0

If x>>1 then

0	0	1	1	1	0

Now the value of x is 14.

If x>>2 then

0	0	0	1	1	1

Now the value of x is 7.