

* Operators

Operator in Java is a symbol that is used to perform operations. for example:
 $+, -, *, /$ etc.

1. Arithmetic

2. Bitwise

3. Relational

4. Logical

5. Assignment

I. Arithmetic Operator

→ Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra.

Operator	Description
$+$ (Addition)	Adds values on either side of operator.
$-$ (Subtraction)	Subtract right-hand operand from left-hand operand.
$*$ (Multiplication)	Multiplies values on either side of operator.
$/$ (Division)	Divides left-hand operand by right by operand.
$\%$ (modulus)	Returns remainder
$++$ (Increment)	Increased the value of operand by 1.
$--$ (Decrement)	Decreases the value of operand by 1.

2. Bitwise Operators

→ Java defines several bitwise operators, which can be applied to the integer types, long, int, short, char and byte.

→ Bitwise operator works on bits and performs bit-by-bit operation.

Operator	Description
& (bitwise and)	Binary AND operator copies a bit to the result if it exists in both operands.
(bitwise or)	Binary OR operator copies a bit if it exists in either operand.
^ (bitwise xor)	Binary XOR operator copies the bit if it set in one operand but not both.
~ (bitwise complement)	Binary ones Complement operator is unary and has the effect of 'flipping' bits.
<< (left shift)	Binary Left Shift operator. The left operand's value is moved by left by the number of bits specified by the right operand.

\gg (right Shift)

Binary Right Shift Operator

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

\ggg (zero fill right shift)

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

3. Relational Operator

→

\sim Operator	Description
$==$ (Equal to)	check if the value of two operands are equal or not, if yes then condition becomes true.
$!=$ (not Equal to)	checks if the value of two operand are equal or not if values are not equal the condition becomes true.
$>$ (Greater than)	checks if the value of ^{left} two operands is greater than the value of right operand.

< (less than)

checks if the value of left operand is less than the value of right operand.

>= (greater than or equal to)

checks if the value of left is greater than or equal to the value of right operand.

<= (less than or equal to)

checks if the value of left operand is less than or equal to the value of right operand.

4. Logical Operators

→

m → Operator	Description
&& (logical and)	if both the operands are non-zero then condition becomes true.
(logical or)	if any of the two operands are non-zero, then Condition becomes true.
! (logical not)	use of reverse the logical state of its of its operand false → true true → false

5. Assignment

\rightarrow Operator	Description
$=$	Simple assignment operator Assign values from right side operand to left side operand.
$+=$	It adds right operand to the left operand and assign the result to the left operand.
$-=$	It subtract right operand from the left operand and assign the result to left operand.
$*=$	It multiplies right operand with the left operand and assign the result to left operand.
$/=$	It divides left operand with the right operand and assign the result to left operand.
$\%=$	It takes modulus using two operands and assign the result to the left operand.

$\ll=$

'left shift AND assignment operator'

$\gg=$

'Right shift AND assignment operator'

$\&=$

'Bitwise AND assignment operator'

$\wedge=$

'Bitwise exclusive OR and assignment operator'

$\mid=$

'Bitwise inclusive OR and assignment operator.'