❖ Operator:-

An operator is a symbol that tells the compiler to perform certain mathematical or logical manipulation.
Java provides a rich operator environment. It provides many types of operators which can be used according to the need. They classified based on the functionality.

➢ Arithmetic Operators:-

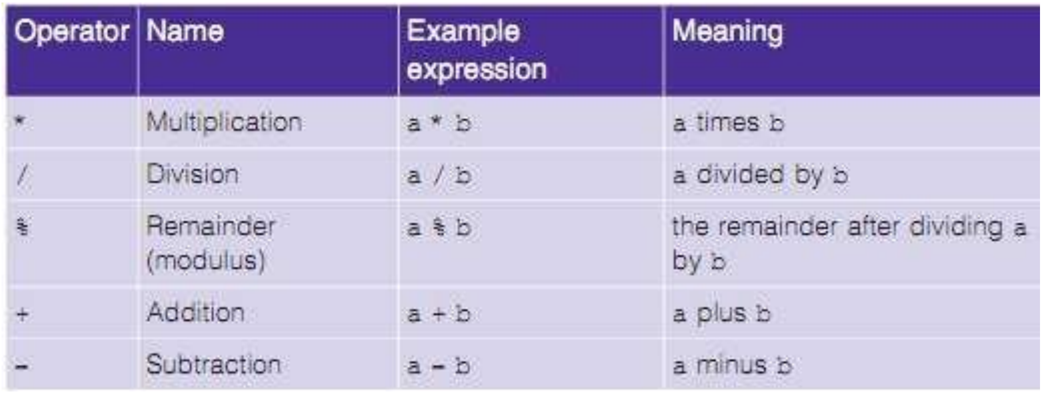
They are used to perform simple arithmetic operation operations on primitive data types. The operands of the arithmetic operators must be of a numeric type. We cannot use them on **boolean** types, but we can use them on **char** types, since the **char** type in Java is, essentially, a subset of **int**.

Syntax:-

Variable/value ***arithmetic_operator*** variable/value;

Ex:-

A*b;   B+5;



Operator	Name	Example expression	Meaning
*	Multiplication	a * b	a times b
/	Division	a / b	a divided by b
%	Remainder (modulus)	a % b	the remainder after dividing a by b
+	Addition	a + b	a plus b
-	Subtraction	a - b	a minus b

➢ Unary Operators:-

Java unary operators are the types that need only one operand to perform any operation like increment, decrement, negation etc. It consist of various arithmetic, logical and other operators that operate on a single operand.

Syntax

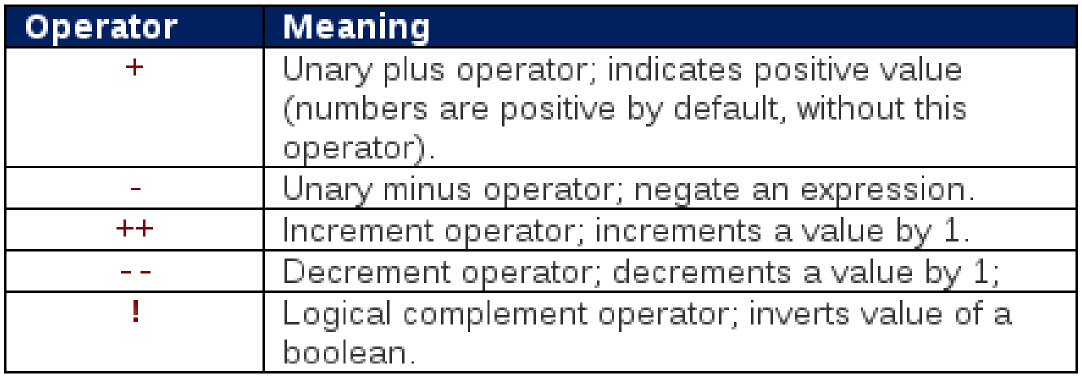
Variable ***unary_operator****;*

(or)

unary_operator Variable;

Ex:-

a++;   --a;



Operator	Meaning
+	Unary plus operator; indicates positive value (numbers are positive by default, without this operator).
-	Unary minus operator; negate an expression.
++	Increment operator; increments a value by 1.
- -	Decrement operator; decrements a value by 1;
!	Logical complement operator; inverts value of a boolean.

- **Assignment Operators:-**

Assignment operator is used to assign a value to any variable.It has right to left associativity. i.e value given on right hand side of operator is assigned to the variable on the left.

Syntax

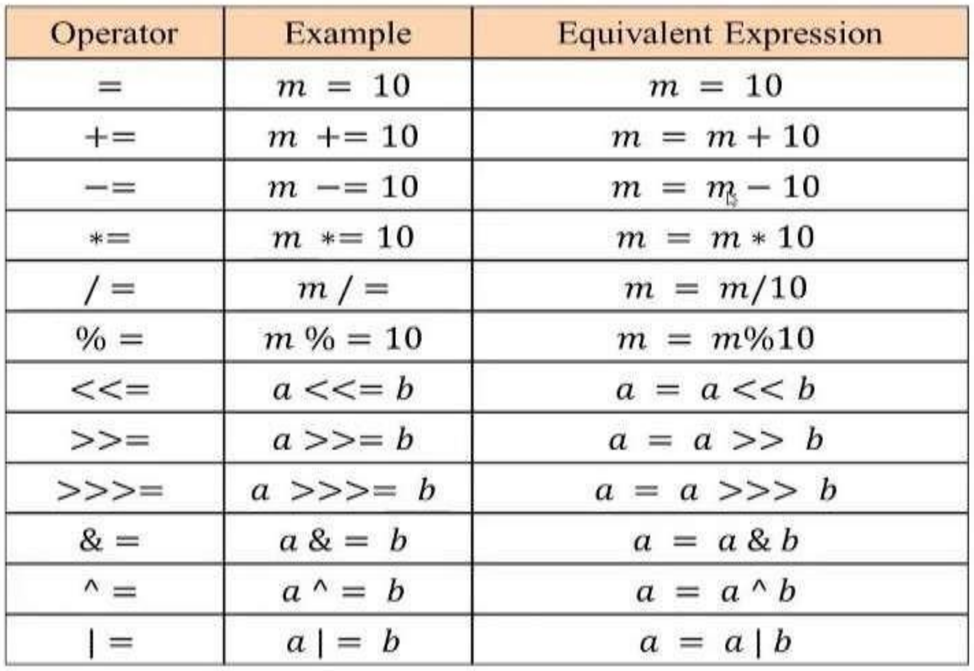
Variable

assignmemt_operatorvalue;

Ex:- a=10;

The assignment operators can be combined with other operators to build a shorter version of the statement called **Compound Statement**.

Ex- a +=10;



Operator	Example	Equivalent Expression
$=$	$m = 10$	$m = 10$
$+=$	$m += 10$	$m = m + 10$
$-=$	$m -= 10$	$m = m - 10$
$*=$	$m *= 10$	$m = m * 10$
$/=$	$m /=$	$m = m/10$
$\% =$	$m \% = 10$	$m = m\%10$
$<<=$	$a <<= b$	$a = a << b$
$>>=$	$a >>= b$	$a = a >> b$
$>>>=$	$a >>>= b$	$a = a >>> b$
$\& =$	$a \& = b$	$a = a \& b$
$\wedge =$	$a \wedge = b$	$a = a \wedge b$
$\mid =$	$a \mid = b$	$a = a \mid b$

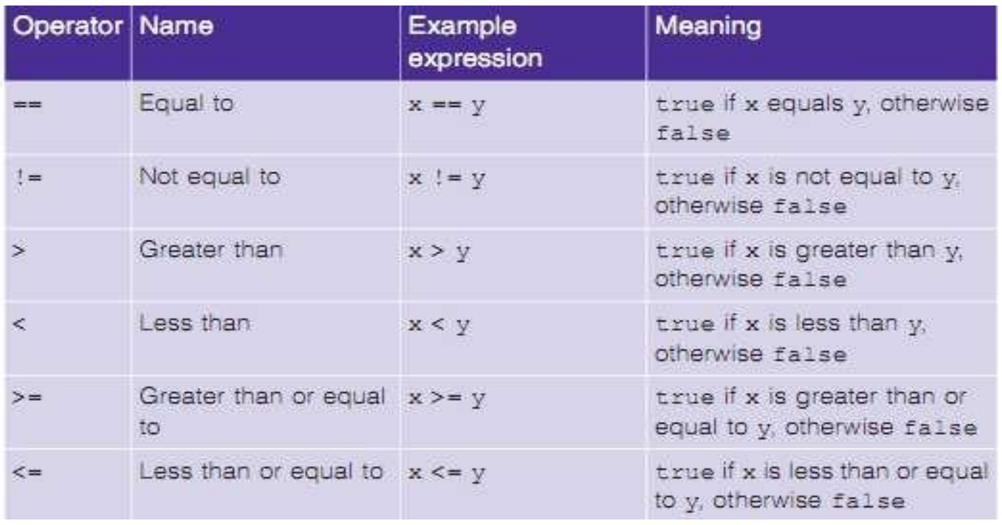
- **Relational Operators:-**

Relational operators are used to check for relations like equality, greater than, less than. They return Boolean value after the comparison.

Syntax:-

Variable/value ***relational_operator*** variable/value;

Ex:-
A==5;
B==A;



Operator	Name	Example expression	Meaning
==	Equal to	x == y	true if x equals y, otherwise false
!=	Not equal to	x != y	true if x is not equal to y, otherwise false
>	Greater than	x > y	true if x is greater than y, otherwise false
<	Less than	x < y	true if x is less than y, otherwise false
>=	Greater than or equal to	x >= y	true if x is greater than or equal to y, otherwise false
<=	Less than or equal to	x <= y	true if x is less than or equal to y, otherwise false

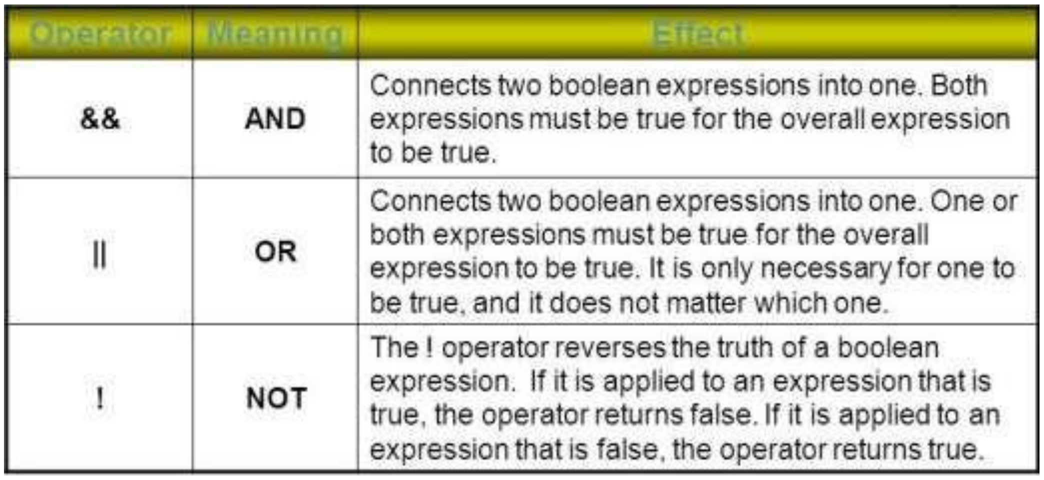
➢ Logical Operators:-

Logical operators are used to determine the logic between variable or value.

Syntax:-

Variable/value ***Logical_operator*** variable/value;

Ex:-
A||b;



Operator	Meaning	Effect
&&	AND	Connects two boolean expressions into one. Both expressions must be true for the overall expression to be true.
||	OR	Connects two boolean expressions into one. One or both expressions must be true for the overall expression to be true. It is only necessary for one to be true, and it does not matter which one.
!	NOT	The ! operator reverses the truth of a boolean expression. If it is applied to an expression that is true, the operator returns false. If it is applied to an expression that is false, the operator returns true.

➢ Ternary Operators:-

Java ternary operator(**?:**) is the only conditional operator that takes three operands. It's a one-linear replacement for if-then-else statement and used a lot in Java programming. We can use the ternary operator in place of if-else conditions or even switch conditions using nested ternary operators.

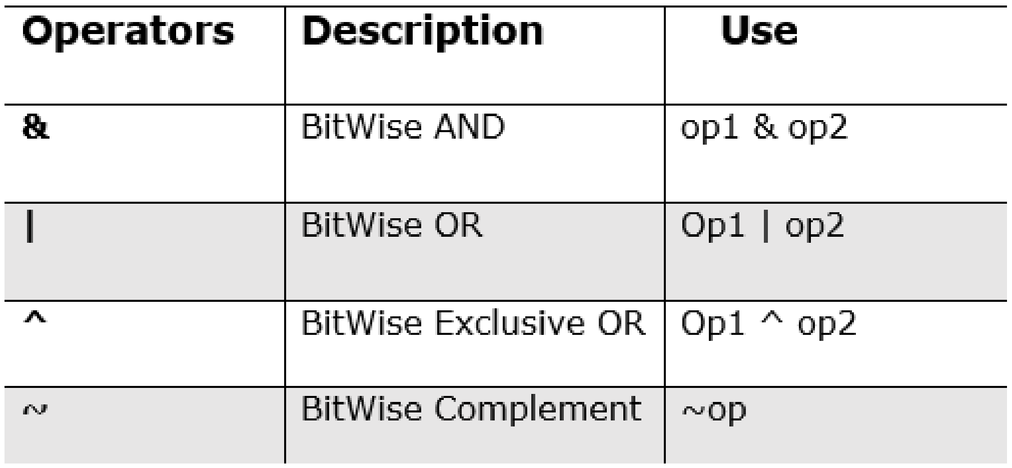
Syntax:-

Condition **?** if true **:** if false

Ex:-
(a>b)?a:b;

➢ Bitwise Operators:-

Bitwise operators are used to perform manipulation of individual bits of a number. They can be used with any of the integral types (char, short, int, etc). They are used when performing update and query operations of Binary indexed tree.



Operators	**Description**	**Use**
&	BitWise AND	op1 & op2
|	BitWise OR	Op1 | op2
^	BitWise Exclusive OR	Op1 ^ op2
~	BitWise Complement	~op

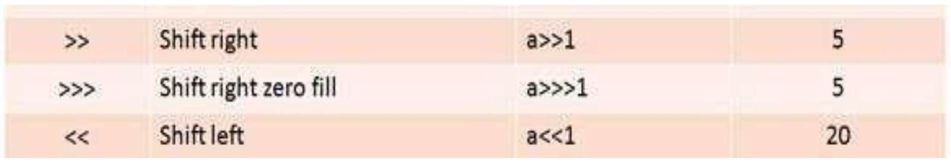
- ## Shift Operators:-

  These 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.

  ### Syntax:-

  Number **shift_op** number_of_places_to_shift;

  Ex:



>>	Shift right	a>>1	5
>>>	Shift right zero fill	a>>>1	5
<<	Shift left	a<<1	20