

Eg: `int a=5, b=7, c=3, d;`

`a = b++ - --c`

`a = 7 - 2`

`b = 8`

`a = 5`

`c = 2`

`d = ++c + b--;`

`d = 3 + 7` `||` `c = 3, b = 6`

`d = 11`

`c = a++ + --b + ++d`

`c = 5 + 5 + 7`

`a = 6, b = 5 + a = 7`

`c = 12`

`a = 6, b = 5, c = 12, d = 19`

Practise Program: →

Eg: `a=5, b=7, c=3, d;`

`a = (b++) - (--c);`

`a = 7 - 2 = a = 5, b = 8, c = 2, d = ?`
`b = 8, c = 2`

`d = (++c) + (b--);` `d = 3 + 8`

`a = 5, c = 3, d = 11, b = 7`

`c = (a++) + (--b) + (++d);`

`c = 5 + 6 + 12`

`a = 6, b = 6, c = 23, d = 12`

Eg: →

$p = \text{---}, b = \text{---}, c = \text{---}, d = \text{---} j$

$p = -1, q = 5, r = 17, S = ?$

① $S = p + p - q - * r;$

$S = -1 - 5 * 17 = -6 * 17$

$p = 0, q = 4, r = 17, S = -102$

$$\begin{array}{r} 16 \\ 2 \\ \hline 112 \end{array}$$

$S = S + 1;$

$S = -103$

$$\begin{array}{r} 103 \\ 17 \\ \hline 086 \end{array}$$

$q = -p + r -;$

$q = -(-103) + (17) = 117$

$p = -104, r = 16, q = 117, S = -103$

$P = q, r = 16, q = 117, S = -103 - 86$

① $S = p + p - q - * r$

$S = (-1) - (5) * 17$

$p = 0, q = 4, r = 17, S = -85$

② $S = S + 1;$

③ $q = -p + r -;$

$q = -(0) + 17$

$S = -86$

$p = 1, r = 16$

$d = 17$

④ $q = -$ $p = 1, r = 16, S = -86, q = 17$

$$\begin{array}{r} 117 \\ 86 \\ \hline 85 \end{array}$$

P	-10	-(01)
q	117	
R	17	
S	-85	-86

∴ Always Follow Bodmas

B → Bracket, O → order, d → division

m → multiplication, a → addition, s → subtraction.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

Concepts

```
int i = 3
```

```
i = i + 1; // i++
```

```
i [3] 3
```

```
print(i); // 3
```

```
int i = 3
```

```
i = ++i;
```

```
i = 4
```

```
print(i);
```

Operator precedence Table of Java. (HW).

Level	Operator	Description	Associativity
16	()	parentheses.	Left to Right.
	[]	array Access	
	new	object creation	
	::	Method Reference	
	.	Member Access	
15	++	unary post-increment	left to right
	--	unary post-decrement	
	+	unary plus	
	-	unary minus	
	!	unary logical NOT	
14	~	unary bitwise NOT	right to left
	++	unary pre-increment	
	--	unary pre-decrement	
13	()	cast	right to left
12	* / %	multiplicative	left to right

11	+ - +	additive String concatenation	left to right
10	<<, >>, >>>	shift	left to right
9	<, <=, >, >=	relational	left to right
8	=, !=	equality	left to right
7	&	bitwise AND	left to right
6	^	bitwise XOR	left to right
5		bitwise OR	left to right
4	&&	logical AND	left to right
3		logical OR	left to right
2	?:	ternary	right to left
1	=, +=, -=, *=, /=, %=, &=, ^=, =, <<=, >>=, >>>=	assignment	right to left
#	Notes: → Increment & Decrement Operator.		

- Increment & Decrement operators are unary operators

Increment Operator (++):

- (i) Pre-Increment Operator.
- (ii) Post-Increment Operator.

(i) Pre-Increment Operator →

Syntax: ++variable-name;

- If a variable is prefixed with increment operator, it is known as pre-increment operator.
- Working: →
First the variable is updated by one value (added) and then the updated value is substituted.

② Post-Increment Operator →

Syntax: variableName++;

- If a variable suffixed with increment operator then it is known as post-increment operator.

• Working →

Here the value in the variable substituted first and then the variable will be immediately added with one value.

Decrement Operator →

②

① Pre-Decrement Operator

② Post-Decrement operator.

① Pre-Decrement operator →