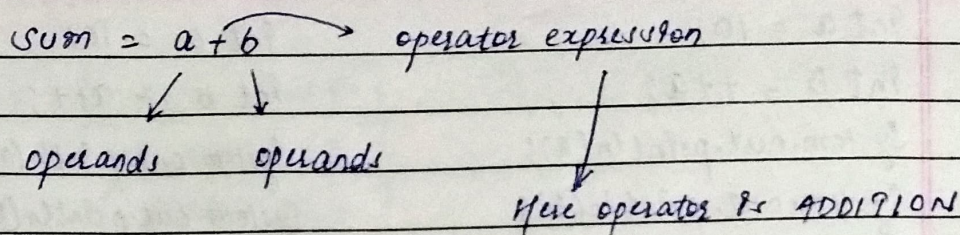


7. Operators



Types of Operator

- ① Arithmetic Operators (Binary/Unary)
- ② Relational "
- ③ Logical "
- ④ Bitwise " (Bit Manipulation)
- ⑤ Assignment "

① Arithmetic Operators

Binary [2 operands]

- $+$ → $a + b$
- $-$ → $a - b$
- \times → $a \times b$
- $/$ → a / b
- $\% \text{ (modulo)}$ → $a \% b$



for remainder

eg $10/5 = 2$

$10 \% 5 = 0$ [Remainder]

$5 \% 3 = 2$ [Remainder]

Unary [Single operands]

- $++$ → $a = a + 1$
- $--$ → $a = a - 1$

Meaning

$a = a + 1 \rightarrow a++$ or $++a$

$a = a - 1 \rightarrow a--$ or $--a$

- $++$ → Increment operator

Pre Increment

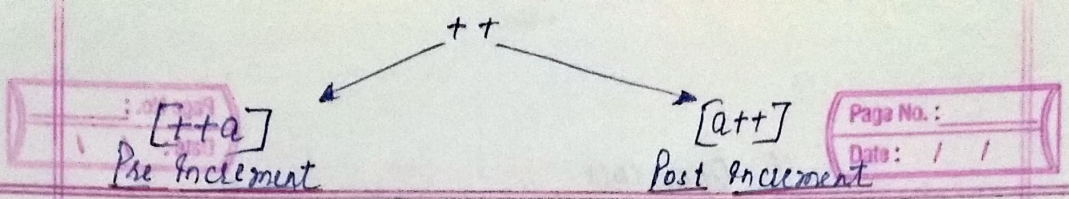
$++a$

1. value change
- then, 2. value use still

Post Increment

$a++$

1. value use
2. value change



```
int a = 10;
int b = ++a;
System.out.println(a);
System.out.println(b);
```

Output: 11
11

```
int a = 10;
int b = a++;
System.out.println(a);
System.out.println(b);
```

Output: 11
10

int a = 10;
int b = ++a;

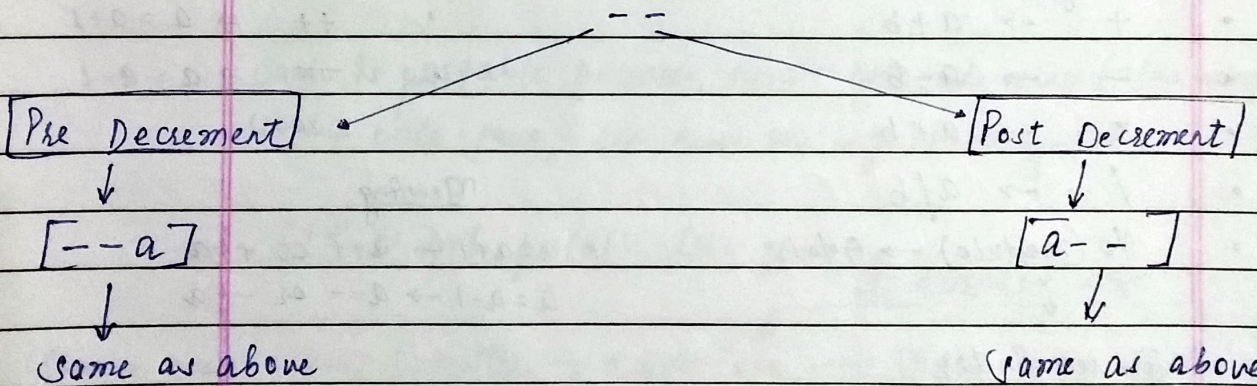
In this whenever the value of 'a' is stored in memory will change first then the same value will assign to b and print

| | |
|----|----|
| a | b |
| 10 | 11 |
| 11 | |

Memory में a = 10 assign है फिर Java 'b' को ~~store~~ करनेगा और उसके अंदर a की जो उसी value को store होगा फिर increment होगा।

int b = a++;
then a increment

- ① पहले 'a' का change होगा
- ② फिर वो value 'b' में store करने के लिए use होगा।



```
int a = 10;
int b = --a;
System.out.println(a);
System.out.println(b);
```

Output: 9
9

```
int a = 10;
int b = a--;
System.out.println(a);
System.out.println(b);
```

Output: 9
10

This operator is used to check if something is equal or not.

Relational Operators

a = 10;
b = 5;

System.out.println(a == b);
Output: false

1. == [equal to equal to]

2. != [not equal to]

3. > [greater than]

4. <

5. >= [Greater than equal to]

6. <=

② not equal to or inequality

a = 10;

b = 5;

cout(a != b);

Output: true

a = 10;

b = 10;

cout(a != b);

Output: false

This operator is opposite of '=='.

③ >

int a = 10;

int b = 5;

cout(a > b);

Output: true

int a = 10;

int b = 5;

cout(b > a);

Output: false

④ Works same as above ③.

⑤ >= [Greater than equal to]

int a = 10;

int b = 10;

cout(a > b);

Output: false

cout(a >= b)

Output: true

Logical Operator

to check logic

① && (Logical AND)

| | Ans | Statement ① | Statement ② |
|-------------------------------------|-----|-------------|-------------|
| System.out.println((3>2) && (5>0)); | T | T | T |
| | F | T | F |
| | F | F | T |
| | F | F | F |

Output: true

System.out.println((3<2) && (5<0));

Output: false

(Stt 1) && (Stt 2)
T T → True

② || (Logical OR)

| | Ans | Stt ① | Stt ② |
|-------------------------------------|-----|-------|-------|
| System.out.println((3<2) (5<0)); | T | T | F |
| | T | F | T |
| | T | T | T |
| | F | F | F |

Output: false

Only false when both statements are false.

③ ! (Logical Not)

Not operator

| Ans | Stt |
|-----|-----|
| F | T |
| T | F |

System.out.println(! (3>2));

Output: false

Assignment Operator

- 1) = $A = B$ [Java assign right value to left]
- 2) +=
- 3) -=
- 4) *=
- 5) /=

2) +=

$$\overbrace{A = A + 10}^{\text{[operand is same]}}$$

↓
or operand should

$$A + = 10$$

same only that

we use this.

3) -=

$$\overbrace{B = B - 5}$$

↓

$$B - = 5$$

- 1 $A = 10;$
- 2 $A = A + 10;$
- 3 $A + = 10;$ // Output is as same as line 2 but java execute it fast (some nano seconds).

And like this other assignment operators also works same.

* Operator Precedence

↓

Check Pdf for table.