Operators

① Arithmatic Operators

+ , - , * , / , % → modulus operator

```
int a = 10;
int b = 20;
int c = a + b;
sysout(c)                    // 30
```

% → remainder

10 % 3 // 1

---

② Unary Operators

1 operand

a) Increment b) decrement c) Negation d) +, -

Increment Operator

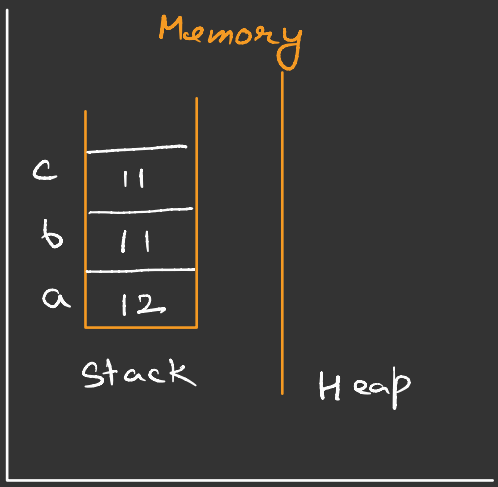
① → Pre increment ++a

② → Post increment a++

① First increment then use the incremented value

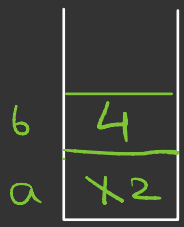
② First use original value then increment

```
int a = 10;
int b = ++a;
syso(b)  // 11
int c = a++;
```

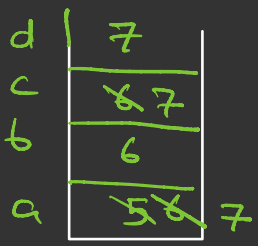


```
int a = 1;
int b = a++ + ++a;
syso(b);
```

1 — 3



```
int a = 5;
int b = ++a;
int c = a++;
int d = ++c;
```



Decrement Operator

- Pre decrement `--a`
- Post decrement `a--`

```
int a = 5
int b = --a

//a = 4
//b = 4
```

c) Nagate (!)

```
boolean a = false

syso(!a) // true
syso(!!a) // false
```

d) +, -

```
int b = +10;
int a = -5;
```

---

(3) Relational Operators

equals, not equals

`>`, `<`, `>=`, `<=` — comparison

`==`, `!=` — equality

```
int a = 5;
int b = 10;

boolean c = a==b;   false
            a>b;    false
            a<=b;   true
            a!=b;   true
```

(4) Logical Operators

And &&

Or ||

a && b

a	b	a && b
T	T	T
F	T	F
F	F	F
T	F	F

a || b

a	b	a || b
T	F	T
T	T	T
F	T	T
F	F	F

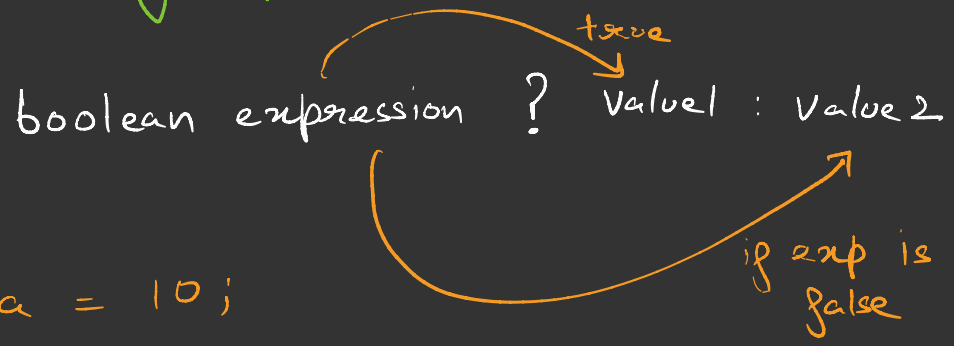
a > b && c > a ⇒ true

a = 10

b = 5

c = 15

(5) Ternary operator ? :




```
int a = 10;
int b = a > 5 ? 3 : 7;
syso( false ? 'a' : 'b' );
```

---

(6) Assignment Operators

=, +=, -=, *=, /=, %=

↳ Put / assign the value of RHS in LHS

```
int a = 2;
int b = a+2    // 4
```

```
a = a + 5;
a += 5;
```