

Operators and Expression

byte b=10;
short s=15;
int i=7;
long l=501;
float f=12.5f;
double d=17.5d;
char c=65;

byte + byte
short + short
byte + short
int + int
short + int

Jab ham in mai sai kisi bhi type par koi operation karenge tab always hama integer value hi milta hai as a result.

Increment / Decrement

post ++, post --
++pre, --pre

Arithmetic

$*, /, \%$ → High precedence

$+, -$ → Low precedence

Bitwise

$\&, |, \sim, ^, \ll, \gg, \ggg$

Relational

$<, <=, >, >=, ==, !=$

Logical

$\&\&, ||, !$

Coercion

int X = b + s int X = c + s;
int X = s + i int X = c + i;
float X = i + f double X = f + d;
float X = l + f double X = l + d;

⊙ Post increment

Ise mai phela value use hota ha then update hota hai.

⊙ Pre increment

Ise mai phela value update hota ha then use hota hai

Operators and Expression

```
int x=5;
x++; - 6
++x; - 7
```

```
float x=5.3;
x++;
6.3
```

```
byte x=4;
x++;
5
```

```
char x='A';
x++; 65 66 'B'
s.o.p(x);
0
```

```
int a=2, b=5, x=4, c
c = a * ++x + b
```

Increment / Decrement

post ++, post --

++pre, --pre

Arithmetic

*, /, %

+, -

Bitwise

&, |, ~, ^, <<, >>, >>>

Relational

<, <=, >, >=, ==, !=

Logical

&&, ||, !

Post increment

```
int x=5, y;
y = x++;
y=5 x=6
```

Pre increment

```
int x=5, y;
y = ++x;
6 6
y=6 x=6
```



```
1 package incdec;
2
3 public class IncDec
4 {
5     public static void main(String[] args)
6     {
7         byte b=5;
8
9         b=b+1;
10
11         System.out.println(b);
12     }
13 }
```

Remember:

b=b+1 & b++ is not same in case of byte. Because jab ham 'b+1' karta ha then 'b' is byte data type and '1' is integer data type then 'b+1' give integer data type but we declare 'b' as byte so we invalid data type error aata hai.

```
compile:
run:
6
BUILD SUCCESSFUL (total time: 2 seconds)
```

Bitwise Operators

AND	\wedge	$\&$
OR	\vee	$ $
NOT	\neg	\sim
XOR	\oplus	\wedge
RIGHT SHIFT		\gg
UNSIGNED RIGHT SHIFT		\ggg
LEFT SHIFT		\ll

S₀ - - d: 0.90x

$\&$

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

$|$

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

\wedge

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Bitwise Operators

int x=10, y=6, z;

x → 00001010

y → 00000110

z = x & y

00000010

2

&

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

!

A	B	A ! B
0	0	0
0	1	1
1	0	1
1	1	1

^

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

$x \rightarrow$ 0 0 0 0 1 0 1 0

$y \rightarrow$ 0 0 0 0 0 1 1 0

$$z = x \mid y \quad \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ & & & & & \underline{8} & \underline{4} & \underline{2} & \underline{1} \end{array}$$

$$z = 14$$

Bitwise Operators

int x=10, y=6, z;

int x=-10;

10 → 00001010

1's comp 11110101

2's comp +1

-10 → 11110110

x → 11110110

x >> 1 → 1011011

x >>> 1 → 0111011

64 32 16 8 4 2 1

123

&

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

|

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

^

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

⊗ Java mai koi bhi '-ve' integer hmasa 2's compliment ka form mai save hota hai.

⊗ Note: 1's compliment ka liya ham phela sabhi value koi uska opp. digit sa replace kar deta hai. Then for 2's compliment add '1' in 1's compliment.