→ Bitwise Operator

AND → &

OR → |

NOT → ~

XOR → ^

(&) AND

a	b	o/p
0	0	0
0	1	0
1	0	0
1	1	1

(|) OR

a	b	o/p
0	0	0
0	1	1
1	0	1
1	1	1

Not (~)

a	o/p
0	1
1	0

XOR

a	b	o/p
0	0	0
0	1	1
1	0	1
1	1	0

Same → 0

diff → 1

→ Left & Right shift operator

↓ "<<" ↓ ">>" $\boxed{2^n}$

↓

int a=2

↓

$\boxed{000 \ldots\ldots 00010}$ → 2

a<<1 → a ko left shift by 1 bit

$\boxed{0000 \ldots\ldots 000100}$ ← add 0 in last bit

↓

4

→ ">>"

5 >>1 $\boxed{\frac{1}{2^n}}$

↳ 5 ko Right shift by 1 bit

$\boxed{0000 \ldots\ldots 0101}$ → 5 $\boxed{5/2 = 2}$

⟶

$\boxed{0000 \ldots\ldots 010}$ → 2

→ Pre / Post Increment / Decrement operator

Pre - Increment → ++a ← Pehle increment karo fir use karo

Post - Increment → a++ ← pehle use karo & fir increment karo

Pre Decrement → --a ← Pehle decrement karo fir use karo

Post Decrement → a-- ← pehle use karo fir decrement karo

for ~~Example~~

```
→ int a=5;                a $\boxed{\cancel{5}6}$
  (++a);
  Cout <<a; → [6]
}

→ main () {
  int a=5;                a $\boxed{\cancel{5}6}$
  Cout << (++a)<< endl; → [6]
```

→ Break & Continue

↓

```
for (int i=0; i<=5; i++) {
    if (i==2){
        break;
    }
    cout <<i<<endl;
}
```

[illegible]

output

```
0
1
```
———— The loop will exit

→ Continue

```
for(int i=0; i<=5; i++) {
    if(i==2){
        Continue;
    }
    Cout << i<< endl;
}
```

o/p ⇒

```
0
1
3
4
5
```

→ Variable Scoping
  ↳ Local Variable
  ↳ Global Variable

```
main() {
    for (int i=0; i<5; i++) {    } Scop of i.
        Cout<<i;
    }
    Cout<<i; ← not accessible
}
```

⇒

```
if (true) {
    int a=202;
    if(true) {                     will access
        int a=303;              → nearest one
        Cout<<"Inside 2" <<a;   → [303]
    }
}
```