

## Bitwise operators

- AND —  $\&$
- OR —  $|$
- NOT —  $!$
- XOR —  $\wedge$

### # AND

$a \& b$

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

→ True

→ False

→ True

→ if condition True then answer is True.

→ if any condition false, then answer false.

Example: upper seen (q. 41)

### # OR

a	b	a   b
0	0	0
0	1	1
1	0	1
1	1	1

→ F

→ T

→ T

→ T

→ if anyone condition (1) True, then otherwise false.

→ Both are condition false then answer false

• NOT ~ Negation

a	not a
0	1
1	0

• XOR

a	b	a ^ b	
0	0	0	- false
0	1	1	- True
1	0	1	→ True
1	1	0	→ false.

→ Find out Two number equal then use xor operator

Example

$$\text{equal} = 2 \wedge 2 = 0$$

→

# Left & Right shift operator

Ex.g.

<< left shift → syntax cout << (a << 1);

>> Right shift → cout << (a >> 1);

<< Left Shift

0000 ——— 001 → a = 1  
                   <<1     ↓ x2 → multiply  
 0000 ——— 0010 → 2  
                   <<1     ↓ x2  
 0000 ——— 00100 → 4  
                   <<1     ↓ x2  
 0000001000 → 8  
                   <<1     ↓ x2  
 0000 ——— 010000 → 16



- left shift in (-) negative number input, then output garbage value return.
- left shift multiply by 2

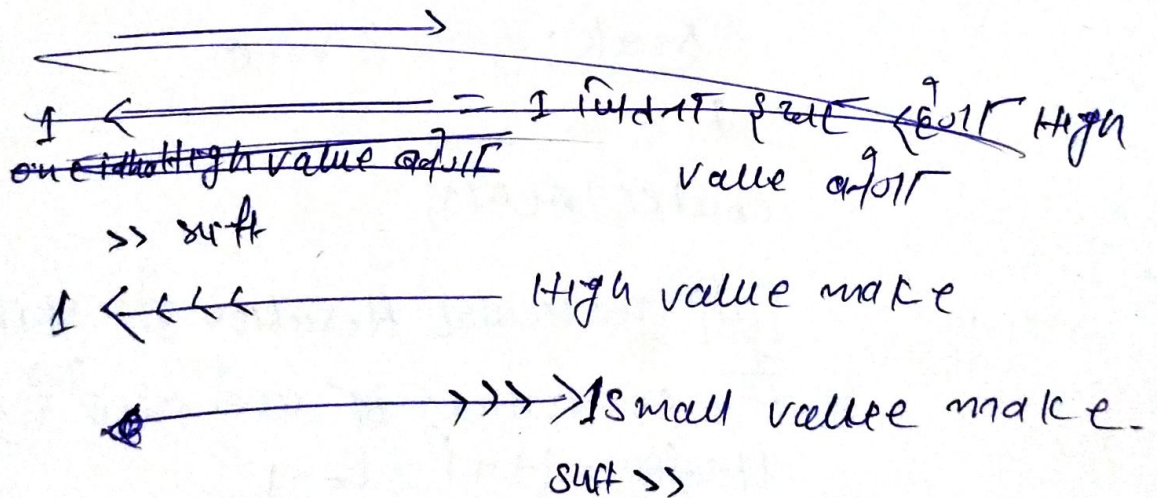
Right shift >>

$a = 8 \rightarrow 0000 \dots 1000$   
 assign  
 $\rightarrow 000000 \dots 1000$

↳ one value delete.  
 ↓  
 4

→ can i say

>> Right shift do then divide by 2 :  
 Right shift not divide by 2 else.







## # Variable Scoping:-

Declaration

int a; → use of variable & declare it  
// initialization → use variable & initialize it

int b = 5;

// updation → use variable & update it  
b = 10;

```
int main() {
```

```
    int b = 5
```

```
    b = 10;
```

```
    if (true) {
```

```
        int b = 15;
```

```
        cout << b;
```

```
    }
```

~~noted~~  
die code

```
    cout << b;  
}
```

→ use main function & variable & use it

in a function same if  
an variable & here  
it will not

## global variable

can be Access any where in the same file.

Bad Practice global variable.

## # Operator Precedence

Practice  $\rightarrow$  Brackets

$$a + b * c / d + e$$

~~$$(a + (b * c / d) + e)$$~~

$$\left( a + \underbrace{(b * c / d)}_{\text{first solve}} + e \right)_{\text{second solve}}$$

## # Switch case:

```
switch (X) {  
    case 1: ---  
        break;  
    case 2: ---  
        break;  
    --  
    default
```

H.W

Switch case.

case 1:  $\rightarrow$  string