# Bit wise operators

| | symbol |
|---|---|
| • AND — | & |
| • OR — | \| |
| • NOT – | ! |
| • XOR – | ^ |

## # AND

a & b

| a | b | o/p | |
|---|---|---|---|
| 0 | 0 | 0 | – True |
| 0 | 1 | 0 | → false |
| 1 | 0 | 0 | → True |
| 1 | 1 | 1 | |

→ दोनों condition True होनी चाहिए, तब Answer होगा True.

→ if any condition false, then Answer false.

Example :- upper seen(देखो)

## # OR

| a | b | a\|b | |
|---|---|---|---|
| 0 | 0 | 0 | → fo |
| 0 | 1 | 1 | — T |
| 1 | 0 | 1 | — T |
| 1 | 1 | 1 | — 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 numbel equal then use XOR oberotars

Example
equal= 2^2 = 0

→

# left & Right shift oberator

Ex.g.

<< left swift → syntax cout<< (a<<1);

>> Right swift → cout<< (a>>1);

<< left shift.

0000.——.001 → a=1

<<1 ↓×2 → multibly

0000——0010 → 2

<<1 ↓*2

0000---00100 → 4

<<1 ↓*2

00000 01000 → 8

↓×2

0000——010000 → 16

→ left shift in (−) negative number input, then output garbage value return.

→ left shift multiply by 2"

Right shift >>

a=8 → 0000----1000
↳ one value delete.

a>>1 → 000000---100
↓
4

→ can i say
>> Right shift do then divide by 2 :
Right shift करते है divide by 2 होता है।

~~1 ← = 1 जितना इधर होगा~~ होगा High value करेगा

~~one [illegible] High value करेगा~~

>> shift

1 ←<<< High value make

——>>>> 1 small value make.
shift >>

# Pre / Post increment / Decrement Operator

Post:- a++ → पहले use ~~print~~ करते then increment

Pre:- ++a → पहले increment करते then use करते और print

- A-- → पहले decrement after then use.
- --A → पहले use करते after then decrement.

# Break and continue Keyword:-

Break:- जब हम loop में होते हैं Break statement लिखा जाता है तो आप उस loop के Body से हम लोग निकल के बाहर आ जाते हैं।

```
for (int i=0; i<n; i++){
    cout<<"Sarfaraj";
     Break;
   }
  cout<<"ALAM;
```

continue:- किसी particular iteration को skip करना हो तो continue; का use करते हैं।
iteration = (++i, i--)

Joke

continue like ताक जाए तो Break like छुटा जाए

# Variable Scoping:—

Declaration

```
int a; → यह पे variable को declare कर दिया
// initialization → यह variable को initialize कर दिया
int b = 5;

// updation → यह variable को update कर दिया
b = 10;


int main(){

        int b = 5
          b = 10;
        if(true){
           int b = 15;
           cout<< b;        die code
        }

        cout<< b; → यह main function के
}                   variable को use करेंगे तो
                    inner function same नाम
                    का variable को [illegible]
                    हो जाएगा
```

global variable

can be Access any where in the same file.

Bad Practice global variable.

# Operator Precedence

Practice → Brackets

a+b*c/d+e

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

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

# Switch case:

```
switch (x) {

    case 1: ---
      Break;

    case 2: ---
        Break;

         --
    default
```

H.W

Switch case.

case 1: $\xrightarrow{'a'}$ string