

17/09/23

Page No.

## Lecture-05

- Bitwise operators
- operator precedence & variable scope
- For loop

### Operator :- Bitwise operator :-

AND  $\rightarrow \&$

OR  $\rightarrow |$

NOT  $\rightarrow \sim$

XOR  $\rightarrow \wedge$

AND  $\rightarrow \&$   $\rightarrow a=2, b=3$

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

2  $\rightarrow 10$

3  $\rightarrow 11$

1 0

2

both bit should be 1 in and (&).

a=5  $\rightarrow 101$

b=7  $\rightarrow 111$

101

x y z

0 0 0

0 1 1

1 0 1

1 1 1

OR  $\rightarrow |$   $\rightarrow a=2 \rightarrow 10$

b=4  $\rightarrow 100$

110

any one bit should be 1 then o/p is 1.

6A



ex<sup>o</sup> a = 3 , b = 6

$$\begin{array}{r} 11 \\ 100 \\ \underline{101} \end{array} \quad \text{5th}$$

(\*) NOT :- ~

X	Z
0	1
1	0

a = 2 → 10

→ 0000 ---- 10

~a ⇒ 111111 ---- 01

negative

take 2's complement for print.

(\*) XOR :- ^

a = 2 →

010

b = 4 →

100

a ^ b →

110

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

ex<sup>o</sup> a = 5 → 101

b = 7 → 111

a ^ b → 010 A

Take 0 for same input.

= left-shift / Right shift :-

• left shift :-

5 << 1

small no.  
multiply

00000 --- 101

0000 --- 1010

3 << 2

→ 000000 --- 111

12.

0000 --- 1100 A

12



• 010 --- 0107

$\ll 1$

greater no. left  
give negative number

100000 --- 0100

negative

$\ll \gg$

• Right Shift :-

15 >> 1

padding 0

starting with zero

Divide  
in right  
shift

5 >> 2

000 --- 0101

small  
no

000 --- 00001

In negative no. padding is compiler dependent

# LOOP :-

increment  $\rightarrow i = i + 1 \rightarrow i++ \rightarrow ++i$   
decrement  $\rightarrow i = i - 1 \rightarrow i-- \rightarrow --i$

post increment

pre increment

Post Decrement

Pre Decrement

= Post increment :-

$i = i++$

int i = 3, a = 2  
sum = a + (i++)

sum = 5

i = 4

int i = 4

int a = i++

a = 4

i = 5



= Post-increment :-

```
int i = 11;
int a = ++i;
```

i = 12  
a = 12

```
a = 2    i = 3
sum = a + (++i)
```

sum = 6  
i = 4

Decrement :- same as increment.

= for-loop :-

for (            ;            ;            )

initialization  
of value

condition / 1 or more

increment-decrement

for ( - ; - ; - )

\* break → It can be used to come outside from the current loop.

\* fibonacci series :-

0, 1, 1, 2, 3, 5, 8, 13, ...

$$n = (n-1) + (n-2);$$

= continue → To skip the particular iteration.