CS & IT ENGINEERING

Data Structures

Introduction to Data Structures

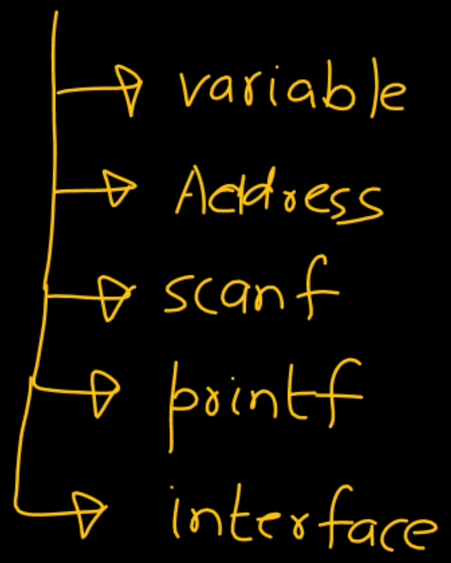
Lec- 03

By- Pankaj Sharma sir

Introduction-3

1,2,3 ✓

doubts ?



Range Calculate
Cyclic property] HW

Operators

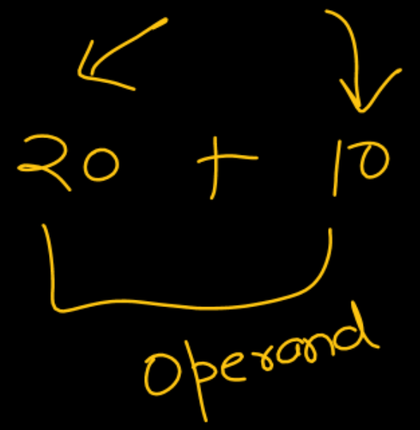
20 + 10 ⇒ 30

20 / 10 ⇒ 2

20 − 10 ⇒ 10

} Arithm. operators

-7 ⇒



2 operand

-7

1 operand

1] Unary

2] binary

3] Ternary

① Assignment operator (=)

binary operator

$x = 10 + 20 \times 3$;

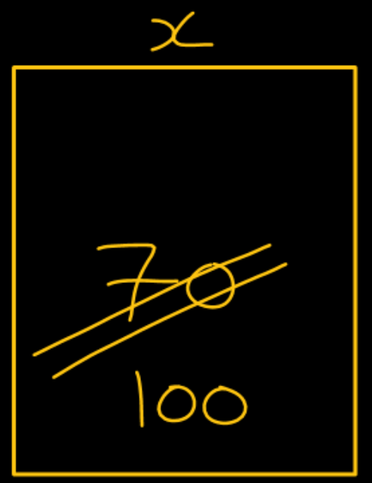
Solve/Evaluate

int x;

x = 10 + 20 x 3 ;

70

x = 100 ;



int x = 2;

100 = x ; ✗

10 + 2 x 3 = x ; ✗

Lvalue = Rvalue ;

int x, y = 4;

x = 10 ; → constant / literal ✓

x = 10 + 2 x 30 / 5 ; → expression ✓

x = y ; → variable ✓

int x = 3;

constant → 2 = x ; ✗

exp → 2 + 3 + 10 = x ; ✗

variable → y = x ;

Arith. Operators

×, /, +, −, %

1) +, − ⇒ unary/binary +12, −12,

binary +, −, ×, / → Math.

a % b ⇒ What is the remainder when a is divided by b

printf("%d", 12 % 5)

⇒ 2

printf("%d", 21%8); → 5

8 | 21 | 2.625
16

50
48

20
16

40
40

X

* both operands must be

int type

```
int x;
x = 13.2 % 5;
```

Udi ke laat Marega

$-5 \% 2$ C Standard $\Rightarrow$

Most of Compilers

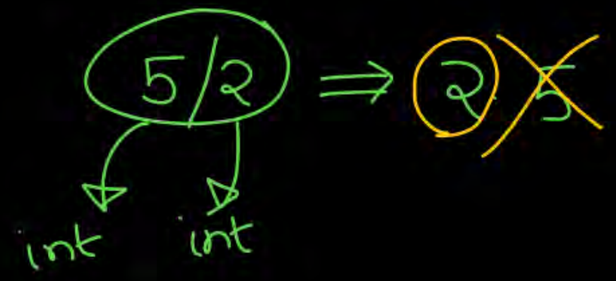
$a \% b \Rightarrow$ sign of a

$-5 \% 2 \Rightarrow$ -ve

$5 \% -2 \Rightarrow$ +ve

$-5 \% -2 \Rightarrow$ -ve

① result of operator ⇒ behaviour of operands



7/2 ⇒ 3

int, int ⇒ int

float, int ⇒ float

int, float ⇒ float

float, float ⇒ float

Result/value

$20 + 30 \Rightarrow 50$

$20 - 30 \Rightarrow -10$

$20 \times 30 \Rightarrow 600$

① %, X, /

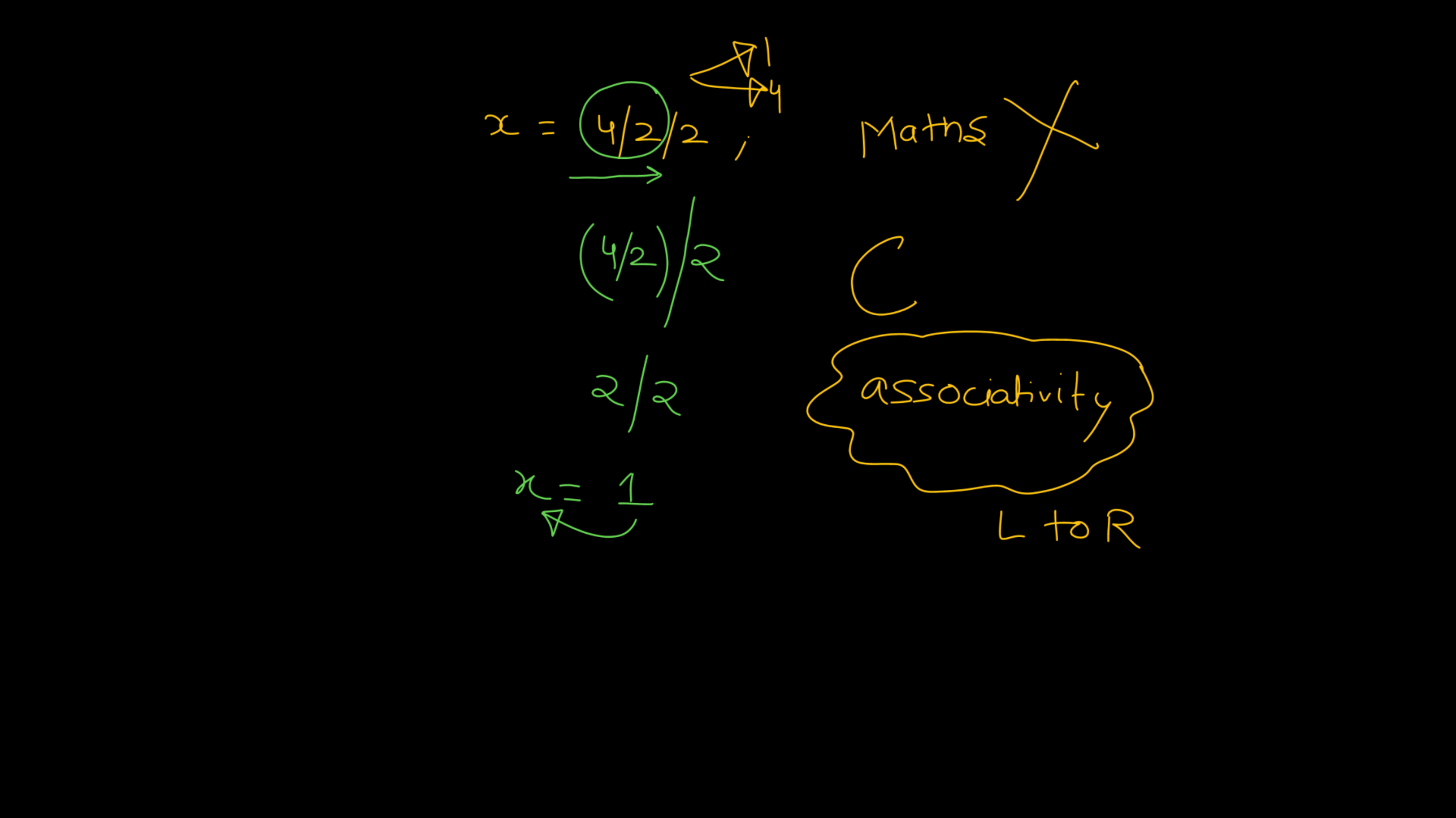
② +, −

```
int x;
x = 2+3x4;
```

$x = 2 + (3 \times 4)$

$2 + 12$

$x = 14$


x = 4/2/2 ;
1
4
Maths
X
(4/2)/2
C
2/2
associativity
x = 1
L to R

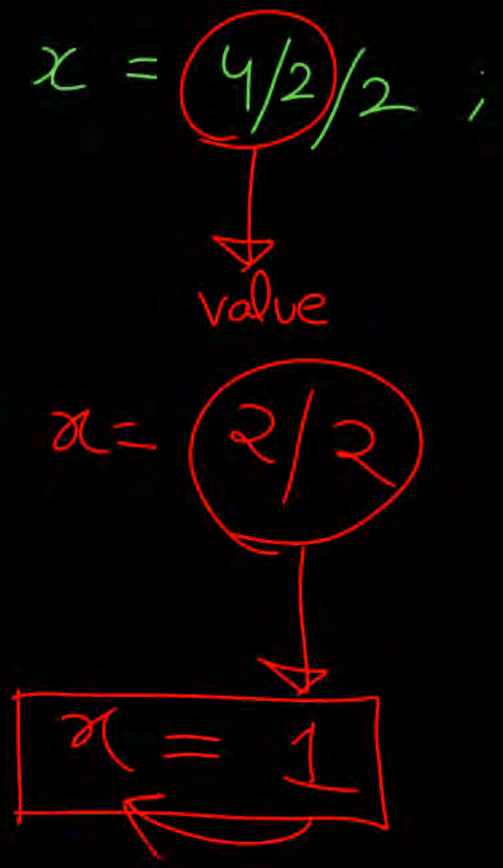
```
int x;
x = 20.2/2 % 3 + 6;
printf("%d", x);
```

① / %
② +
③ =

float 20.2/2 int

float 10.1 % 3 + 6 ;

Error



_____ = operator

```
void main(){
    int x ;
    printf("%d", x=10);
}
```

10

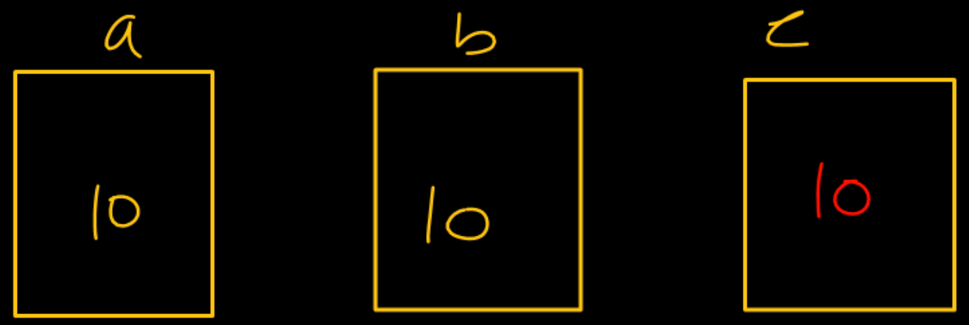
O/P : 10

= (R to L)

```
int a, b, c ;
```

```
a = b = c = 10;
```

① ② ③

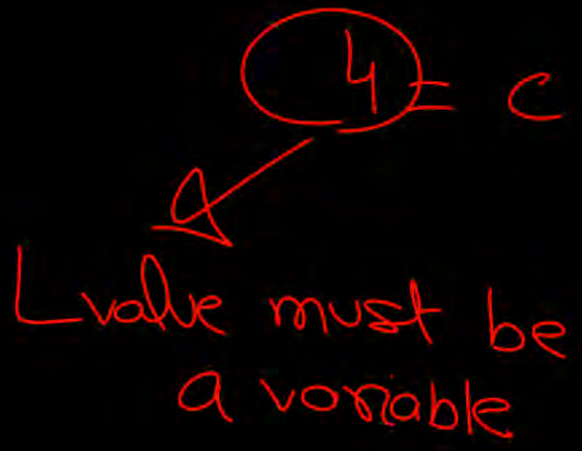


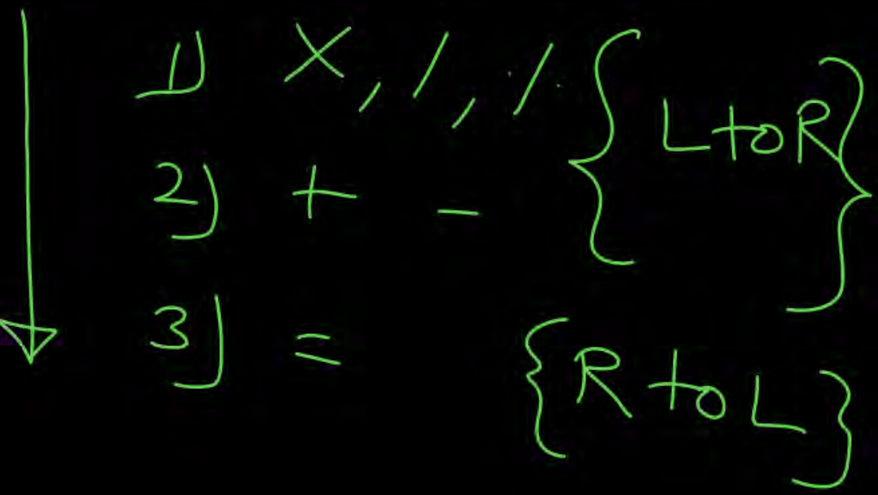
c = 10 → What is the value of this operation?

a = b = 10

a = 10

```
int a,b,c;
a=b=4=c;
printf("%d%d%d",a,b,c);
```





Relational Operators

value/Result

$10 < 20 \Rightarrow$ Is 10 less than 20 ? → True / False

1

```
printf("%d", 10<20);    1
```

$20 < 10 \Rightarrow$ False $\Rightarrow$ value/result $\Rightarrow 0$

```
printf("%d", 20<10); ⇒ 0
```

1] < ✓
2] > ✓
3] <=
4] >=
5] ==
6] !=

a<=b ⟹ 10<=10 → Is 10 less than 10 / Is 10 Equals to 10 ✓

① printf("%d", 10<=20); 1
② printf("%d", 10>=20); 0

10==20 Is 10 Equals to 20 ? O/P: 0

③ printf("%d", 10==20); 0
④ printf("%d", 10==10); 1

10 != 20 Is 10 not equals 20 → Yes / No

10 not equals 20 → T / F

printf("%d", 10 != 20); 1

printf("%d", 10 != 10); 0

Result of every relational operator is either 0 or 1.

High

lowest

1) x, /, %

2) +, −

3) <, <=, >, >=

4) ==, !=

5) =

{ L to R }

{ L to R }

R to L

```
int x;
x = 10+3<1<0<-1>2-6;
printf("%d",x);
```

1

① +, −

② <, >

③ =

$x = 13 < 1 < 0 < -1 > 2-6;$

$x = \underset{\text{False}}{13<1} < 0 < -1 > -4$

$x = \underset{\text{False}}{0<0} < -1 > -4$

$x = \underset{\text{False}}{0 < -1} > -4$

$x = 0 > -4$

$x = \underset{\text{True}}{0 > -4}$

$x = 1$

```
int i ;
i=3 ;
```
✓ valid

```
int i ;
i = printf("Pankaj");
```
6

printf ⇒ No. of symbols printed by it

```
Pankaj
```

```
printf("%d", 2+3*6/2);
```

① Evaluate

2 + 18/2

2 + 9

11

```
printf("%d", 11);
```

1

printf("%d", ~~printf("%d", 3+4/2)~~)

Evaluate

printf("%d", 1);

printf("%d", (3+4/2))

3+2

5

51

printf("%d", printf("%d", 3+4/2))

value

1

printf("%d", 1)

printf("%d", 3+4/2);

5

51

Logical Operators

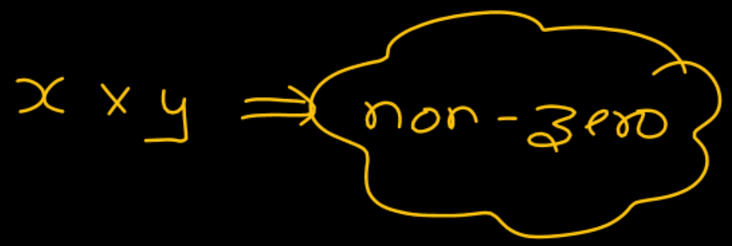
non-zero ⇒ True

zero ⇒ Fdse

(i) Logical AND → और

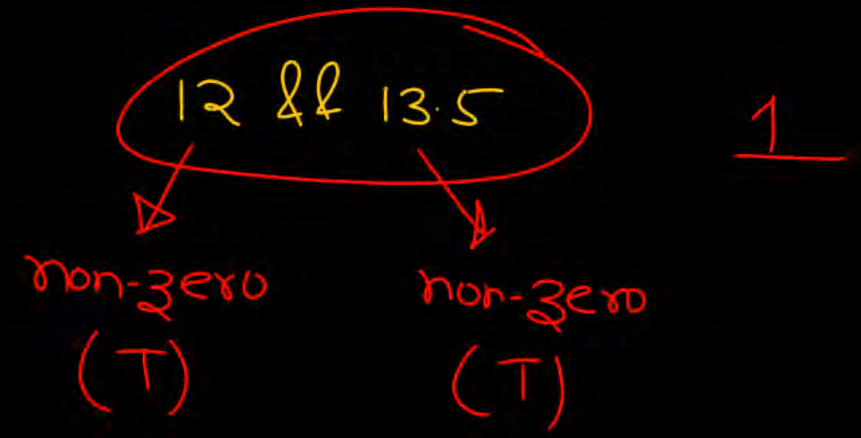
x AND y

x	y	x AND y	Result/value
F	F	F	0
F	T	F	0
T	F	F	0
T	T	T	1


x x y
non-zero

Result
0, 1

Logical AND (&&)



1

```
printf("%d", 2 && 3.6);   1
printf("%d", 2.1 && 3.8); 1
printf("%d", 2.1 && 0);   0
printf("%d", 0 && 2.1);   0
```

```
void main(){
    int i;
    i = printf("Hello") && printf("sir");
    printf("%d", i);
}
```

1:

5 3

1

i = 5 && 3 ,

Hellosir1

Logical OR (||)

→ choice (अथवा)

If atleast one operand is non-zero ⇒ 1

If both are zero (False) ⇒ 0

a	b	a || b	Result/value
F	F	F	0
F	T	T	1
T	F	T	1
T	T	T	1

1. `printf("%d", 21||3);` 1
2. `printf("%d", 2.1||3.8);` 1
3. `printf("%d", 2.1||0);` 1
4. `printf("%d", 0||0.0);` 0

Logical NOT (!)

!non-zero ⇒ 0
!zero = 1

NOT(True) = False

NOT(False) = True

!5 = !non-zero = !True = False = 0

!non-zero = 0

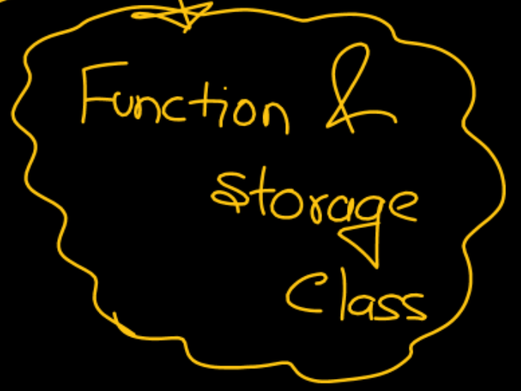
!0 = !zero = !False = True = 1

4,5,6

Friday

2nd chapter

sat sunday




THANK YOU GW
SOLDIERS !