

CS & IT ENGINEERING



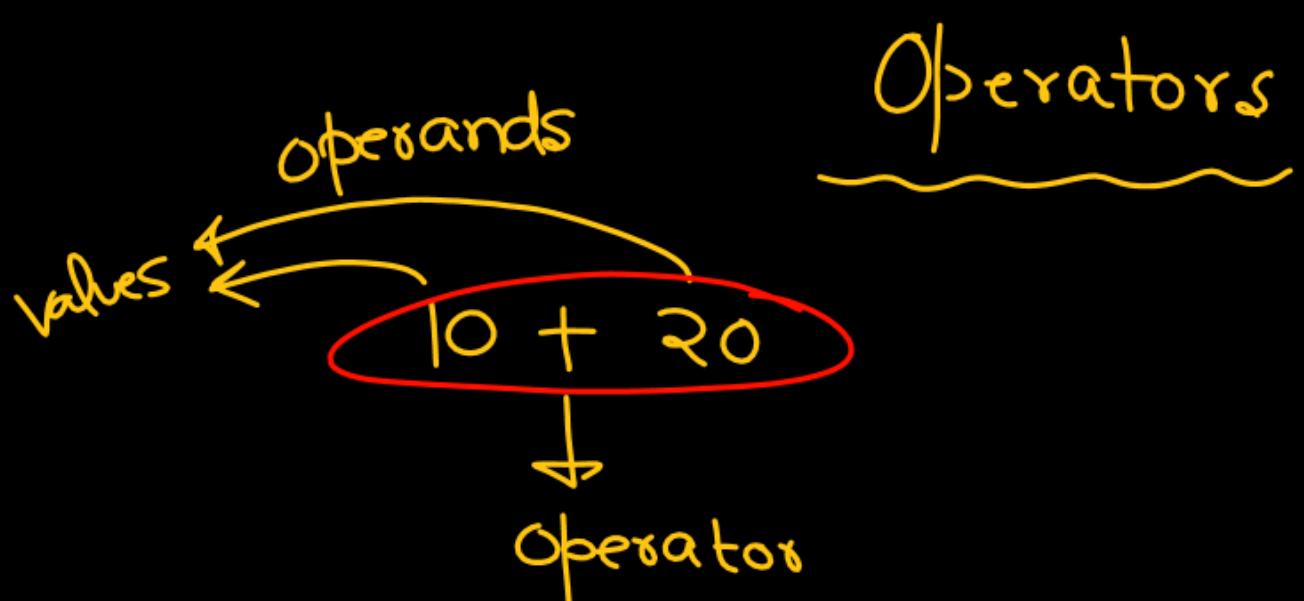
Programming in C
Chapter -1
Data types and Operators
Lec- 04



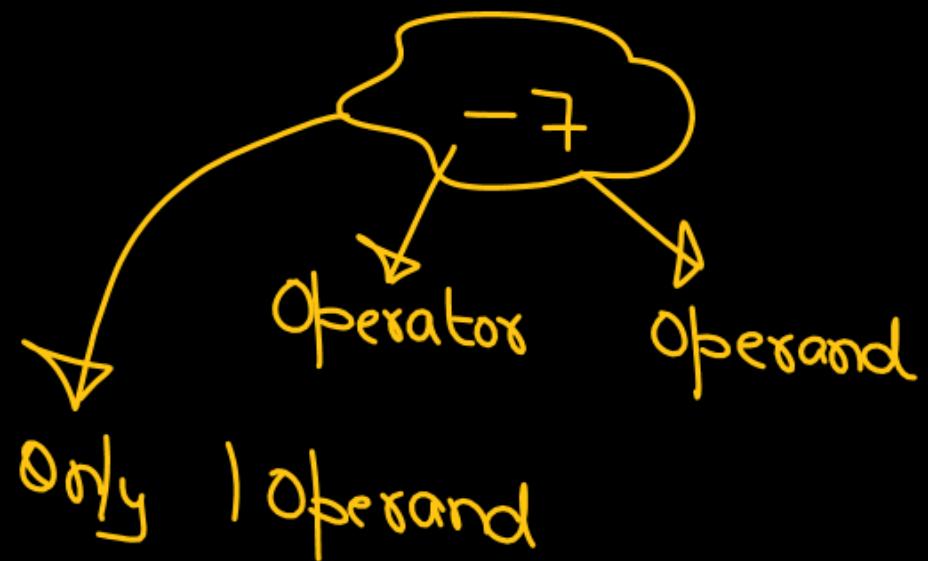
By- Pankaj Sharma sir

TOPICS TO BE COVERED

Operators-I



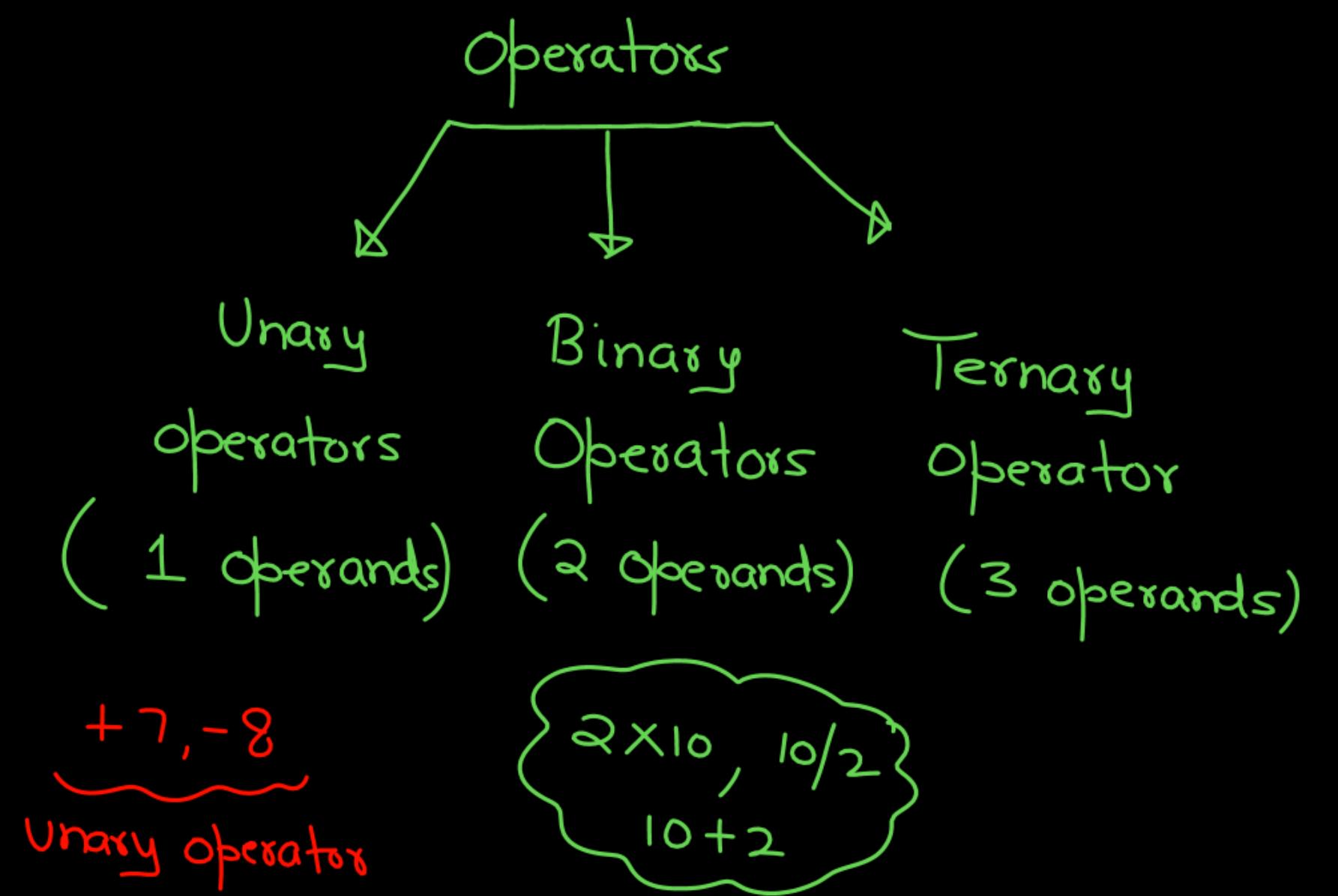
2 operands (values)



(i) ~~ER~~ operation on
 \Rightarrow Result / output / value

Diagram showing the result of the unary operation:

- A red oval contains the number -7 .
- An arrow points from this oval to the number -7 written in red.



① Assignment (=) Operator

Math

$$a = 3 \times 2 + 5$$

Evaluate

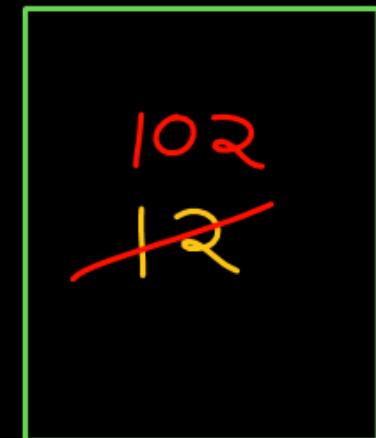
int a ; declaration
int a=12; dec + initial value
a

```
#include<stdio.h>
void main(){
```

```
    int a=12;
```

a = 102 ;

↓
assignment
operator



1024

LHS = RHS

[int a=20;
10] = a;] ➔ Invalid
↓
Constant / literal integer



Q + 3

② Expression :

① Statement :

→ Sentence

→ Statement

Expression:

{Statement having some
value}

Compile

```
#include<stdio.h>
Void main(){
    20 + 10;
}
```

void main(){
 int a{};
 a = 20{};

≡

}

✓

value → expression → Statement

declarative statement

Assignment statement

Lvalue can not
be a constant

1)

Lvalue = Rvalue

int a;
10 = a;] invalid

Lvalue
can not be
an expression

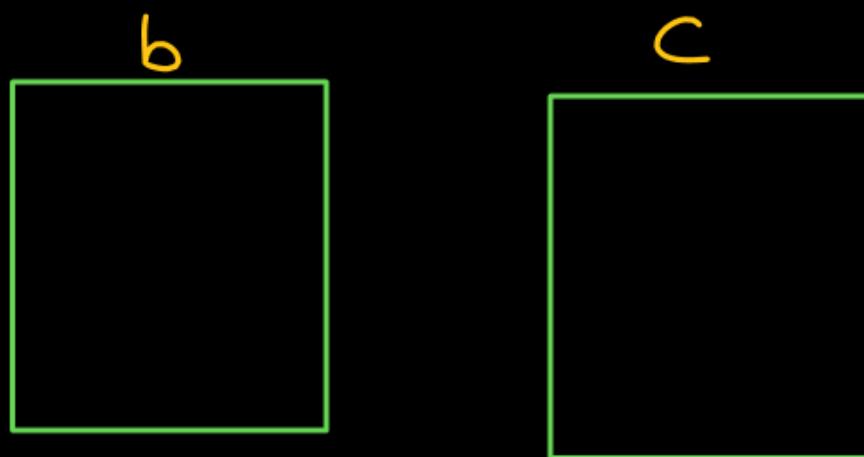
2)
int b, c;
b + c = 100;] invalid

Lvalue can be
variable

3)
a = 10;

Lvalue

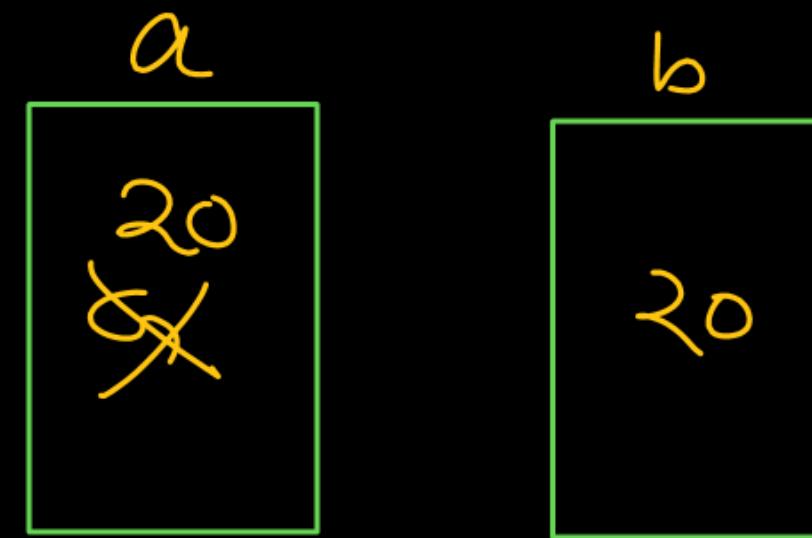
- exp X
- constant X
- variable ✓



1) $a = 10;$
 constant/
 literal

- (i) binary operator
- (ii) Lvalue = Rvalue
(variable) variable, literal,
 expression

2) $\int \text{int } a;$
 $a = (3+4);$
 Rvalue can be exp.



3) $\int \text{int } a, b = 20;$
 $a = b;$
 Rvalue can be var.
 20

$\left[\begin{array}{l} \text{int } a; \\ = a; \end{array} \right]$ ud ke laat
 Marega

Arithmatic Operator

{ + , - } \Rightarrow

Unary +
Unary -

Binary

+ , - , * , / , %

Math

Modulus operator

ud re float
Marega

Compiler

int a;
a = ;

→ Rules

Lvalue = Rvalue;

Math

$$8/2 \times 4 \Rightarrow \text{Evaluate}$$



- , + , × , ÷ , %

Modulas (%): $a \% b$: { What is the remainder
when a is divided by b }

→ binary

7 % 3

$$3 \overline{)7} \quad 2$$

$\frac{6}{\text{Rem}}$

→ Result/o/p/value 1 ← Rem

printf(" %.d", ~~7 % 3~~¹); 1

12 / 5

$$5 \overline{)12} \quad 2 \cdot 4$$

float X

(ii) % : Both operands for modulus operator must be integer type

Otherwise → Tiger

→ Vidyut Jamwal

$$2 \% 7$$

O/P
 2 $\overline{)2}^0$
 0
 2 ←

$$4 \% 5$$

4 $\overline{)4}^0$
 0
 4 ←

$a \% b$: The sign of the result
 is same as the sign of
 first operand.

$-12 \% 5$: -ve

$12 \% -5$: +ve

$-12 \% -5$: -ve

int a;

$a = 3 \times 2 + 1 ; ;$

↓
Evaluate ?

$a = 6 + 1 ;$

$a = 7 ;$

a
7

Priority / Precedance

High { ×, /, % }
↓
lowest =

int a;

$$a = \frac{8}{4} \times 2 ;$$

$\xrightarrow{\quad}$

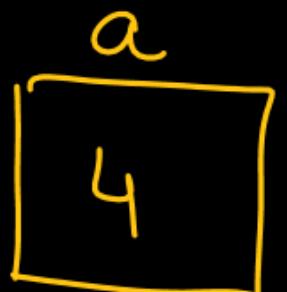
$\times, /, \%$

$+, -$

$=$

$$a = 2 \times 2 ;$$

$$a = 4 ;$$



Associativity

Left to Right

5 (i) `printf("/d", 2+3);`

Result ↗

(i) Eval. $2+3$
5

Evaluate = 5

17 (i) `printf("/d", 2+3*5);`

(i) Eval. $2+3\times 5$

$\downarrow \times, /, \%$
 $+,-$

(ii) `printf("/d", 17);`

17

= is also an operator

Result/Value	Op. का	value O/P
$2+3$		5
2×3		6
$12/3$		4
$12 \cdot 3$		0

int a;

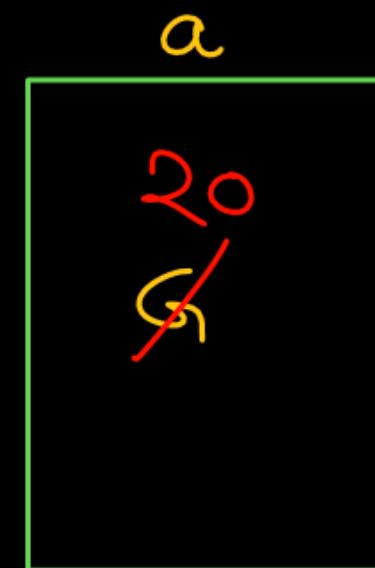
~~printf("%d", a = 20);~~

(i) $a = 20$

assign 20
to a

(ii) ~~printf("%d", 20)~~

20



```
int a ;
```

```
a = 10 ① × 20 ② × 2 ;
```



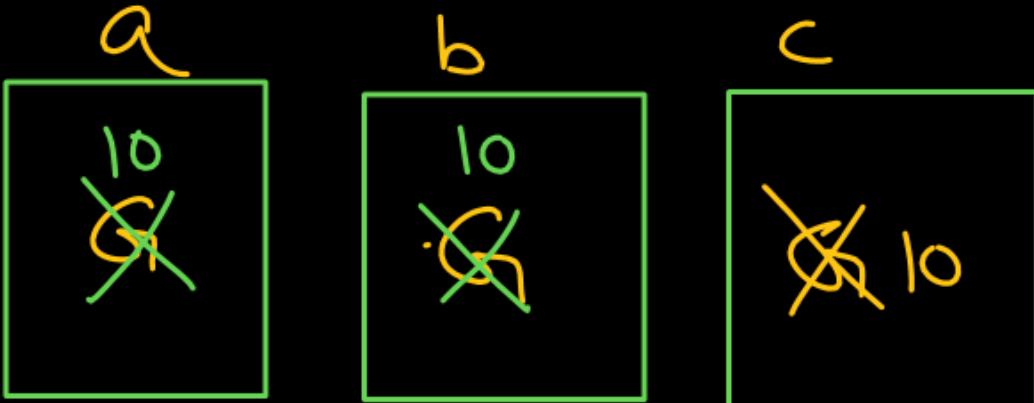
```
int a, b, c ;
```

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

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

=

Right to Left



(i) Assign
10 to *c*

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

(ii) Assign
10 to *b*

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

```
int a, b, c;  
a = b = 10 = c;
```

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

$a = b = \boxed{10 = c};$

उद के लाल मरेगा
विजय
विजय

Lvalue
can not
be
literal/constant

Result of an operator :

depends on
Operands

(i) $12 / 5$

\downarrow \downarrow
int int

~~2.4~~ Result
2

(ii) $12.0 / 5$

\downarrow \downarrow
float int

2.4 2.4

(iii) $12 / 5.0$

int \downarrow
 float

2.4

(iv) $12.0 / 5.0$

2.4

int,int int
int,float float
float,int float
float,float float

```
int a;  
a = 8.0 * 3 / 5 + 1;  
printf("%d", a);
```

Anna

24 Ghante

Chaukanna

$a = 8.0 \times 3 / 5 + 1$;

floating type

$a = 24.0 \% 5 + 1$;

Compiler: Udike loot Marwadi

$\times \quad / \quad + \quad =$

Relational Operators (binary)

(i) <

10 < 20 : Is 10 less than 20 ?

(ii) <=

Statement is true

(iii) >

O/P : 1

(iv) >=

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

(v) ==

1

(vi) !

$40 < 20$: Is 40 less than 20 ?

Statement is False

O/P : 0

`printf("%d", 40 < 20);` 0

(i) Every relational Operator is a binary operator.

(ii) The result of relational Operators is either 0 or 1 .

\leq
 \leq

$a \leq b$

Either $a < b$
or
 a is equal to b

$|0 \leq 5$

<code>printf("%d", 0 <= 20);</code>	1
<code>printf("%d", 0 <= 5);</code>	0
<code>printf("%d", 0 <= 10);</code>	1

```
printf("./d", 10>=20); 0  
printf("./d", 20>=20); 1  
printf("./d", 30>=5); 1
```

$L \xrightarrow{f} R$ {
·, ×, /
+, -
<, <=, >, >=}

printf ("./d", a = 10 >= b = 2);

$a == b$: Is the value of a same as the value of b ?

$10 == 10$: Is 10 equals to 10 ?

True

O/P : 1

```
printf("%d", 10==10); 1
```

```
printf("%d", 10==20); 0
```

$a != b$: Is the value of a not equals to the value of b ?

$10 != 10$: Is 10 not equals to 10

False

```
printf("./d", 2 != 2); 0  
printf("./d", 2 != 0); 1
```

Binary Arith.
High

①

X, /, %

②

+, -

③

<, <=, >, >=

④

==, !=

Low
⑤ =

Weekend

L to R

Right to Left

