

## (#) Defining Symbolic constants :-

↪ we need some constants in a program repeatedly.

$$\text{Ex:- } \pi = 3.1415$$

Syntax:-

#define Symbolic-name value

↪ Preprocessor directive

#define Pi 3.1415

Definition  
Section ↪

A = Pi \* r \* r;

⇒

A = 3.1415 \* r \* r;

---

Q. WAP to find the area of circle

```
#include < stdio.h >
```

```
#define Pi 3.1415
```

```
int main()
```

```
{
```

```
float area, radius;
```

```
printf("Enter the radius");
```

```
scanf("%f", & radius);
```

```
area = Pi * radius * radius;
```

```
    printf("Area = %f", area);  
    return 0;  
}
```

## Operator and Expression

Expression :- A formula in which Operands are linked to each other by the use of operators

$x = a + b * c - d / e$  expression

; a, b, c, d, e, x are operands

; =, +, \*, -, / are operators

## Types of Operators in C

### ① Arithmetic operators

addition $a+b$	operator +	Meaning addition or unary plus
subtraction $a-b$	-	subtraction or unary minus
multiplication $a \times b$	*	multiplication
division $a/b$	/	division
Remainder or modulo division $a \% b$	%	remainder or modulo division

$$\Rightarrow 4 \% 2 \quad 5 \% 2$$

$$\Rightarrow 0 \quad \Rightarrow 1$$

## ② increment /decrement operator

Increment

→ pre increment  $\Rightarrow ++a$  ( $a=a+1$ )

→ post increment  $\Rightarrow a++$  ( $a=a+1$ )

Ex:

$$b=5;$$

$$a=++b;$$
  


$$a=6, b=6;$$

$$b=5;$$

$$a=b++;$$
  


$$a=5, b=6;$$

## ③ Assignment Operator

operator	example	meaning
=	$a=b$	$a = \underbrace{b}$
+=	$a+=b$	$a = a+b$
-=	$a-=b$	$a = a-b$
*=	$a*=b$	$a = a*b$
/=	$a/=b$	$a = a/b$
%=	$a\%b$	$a = a\%b$

## ④ Relational Operator

↳ if relation is true returns 1  
↳ if " " " false " 0

Operator meaning

$= =$

Equal to

$\neq$

Not equal to

$>$

greater than

$\geq$

greater than or  
equal to

$<$

less than

$\leq$

less than or equal  
to

$$\underbrace{a == b}_{\text{True (1)}} ; \text{let, } a = 5 \\ b = 5$$

$$5 == 5 \} \text{ true}$$

$a > b, 5 > 5 \Rightarrow \text{false (0)}$

## ⑤ logical operator.

$\&$  logical and

$\|$  logical or

! logical NOT

$(C \& C == 5) \&& (d > 5)$

$C = 5$   
 $d = 2$

$5 == 5$        $\&&$        $2 > 5$

↓                  ↓  
True(1)          False(0)

→ False(0)

$(C \& C == 5) || (d > 5)$

↓                  ↓  
True(1)          False(0)

→ True(1)

## ⑥ Size of operator

```
int a;  
printf("size of a is %d",  
      Sizeof(a));
```

Syntax :- Sizeof(a);

## ⑦ comma operator

```
int a, b, c;  
int a, b=5, c;
```

## ⑧ conditional operator

↳ Ternary operator.

Syntax :-

Condition ? true-statement :  
false-statement;

Example

$\text{max} = (a > b) ? a : b ;$

true  
false

$a = 7, b = 5 \rightarrow \text{true}$

$(7 > 5) ?$   
 $\boxed{\text{max} = 7}$

## ⑨ Bitwise operator

Operators

&

|

^

~

<<

>>

Meaning

Bitwise AND

Bitwise OR

Bitwise XOR

~ Complement

<< Shift left

>> Shift right