Operators

Subject: C Language Date: 22/12/2025

1. Arithmetic Instruction:

⇒ An instruction which is used to manipulate data using operators, is known as Arithmetic Instruction.

Example :— 8 + 4 → operator

operands

2. Classification of Operators :—

i) Unary operators +, −, ++, −−, sizeof()
ii) Arithmetic operators *, /, %, +, −
iii) Bitwise operators &, |, ^, ~, >>, <<
iv) Relational operators <, >, <=, >=, ==, !=
v) Logical Operators !, &&, ||
vi) Conditional operator ?:
vii) Assignment operator =, +=, −=, *=, /=, %=

3. Operator Precedence Table (High → Low)

⇒ Priority	Operators	Description
1	()	Parentheses
2	++ −−	Increment/Decrement
3	! sizeof	Unary operators
4	* / %	Arithmetic
5	+ −	Arithmetic
6	< <= > >=	Relational
7	== !=	Equality

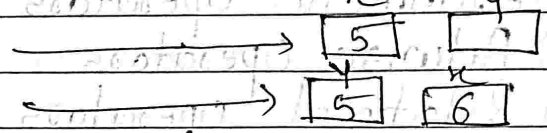
Subject: Date: __/__/__

8	&&	Logical AND
9	||	Logical OR
10	?:	Conditional
11	= += -= *= %=	Assignment
12	,	Comma

NoTE:- Post increment ki priority sabse last me hoti hai.

Example:- Find output of the following program?

```
#include <stdio.h>
int main()
{
    int x=5, y;
    y=x++;
    printf("%d %d", x, y);
}
```

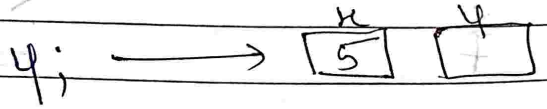


output:- 65

NoTE:- Pre increment ki priority hamesha jyada hota hai remaining operator se.

Example:- Find output of the following program?

```
#include <stdio.h>
int main()
{
    int x=5, y;
```



Subject: Date: /   /

```
    y = ++x;  ──→  x [6]  y [6]
    printf("%d %d", x, y);
}
```

output :— 66

Note:— Increment / Decrement operator hamesha variable ke uppar lagta hai.

4. Sizeof( )

yaha pe hum :—
i) Data type
ii) Variable
iii) Constant
inhi teeno me se kisi ek ko likh sakte hai.

1. Example :— Data type

```
int x;
x = sizeof(float);
printf("%d", x);   // output — 4 bytes

x = sizeof(double);
printf("%d", x);   // output — 8 bytes

x = size of(char);
printf("%d", x);   // output — 1 bytes

x = Sizeof(int);
Printf("%d", x);   // output — 4 bytes
```

Date: __/__/__

Subject: _____

2. Example :— Variable

```
int x, y;
float m;
char ch;
double dl;

x = sizeof(dl); //output — 8 bytes
x = sizeof(ch); //output — 1 byte
x = sizeof(y); //output — 4 bytes
x = sizeof(m); //output — 4 bytes
```

3. Example :— Constant

```
int x;
x = sizeof(35); //output — 4 bytes
x = sizeof(4.7); //output — 8 bytes
x = sizeof('A'); //output — 4 bytes
```

Note :-
- Real Constants are by default of double type.
- Integer Constant is int.
- character Constant is int.
- Real Constant par kabhi modulus apply nahi hota.
  Example :— 3.5 % 2 (wrong)

- char x = 'A' / char x = 65 (Both same)
- int x = 65 / int x = 'A' (Both same)

Subject: Date: / /

5. Some Important point.

i) Integer Constant / Integer Constant ⇒ Always give Integer Constant.

Example :- 5/2 = 2

ii) Integer Constant / Real Constant ⇒ Always give Real Constant.

Example :- 5/2.0 = 2.5

iii) x % y = 0 (It means ki x hamesha divisible hoga y se).

iv) x/5 = 3 (It means ki x hamesha divisible nahi hoga 5 se).

v) x/10 (It means ke value of x without last digit).

vi) x % 10 (It means ki last value of x).

vii) 3 % 4 (It means ki jab bhi aap small no. ko modulus karoge big no. se to hamesha small no. hi result me aayega).

viii) Agar Same priority ke operator ek se jyada lage ho same line me to hum hamesha usse left to right Solve karte hai.

Ex :- 5 > 4 > 3

Subject: Date: / /

6. —: Bitwise Operators :—

i) AND — &
ii) OR — |
iii) XOR — ^
iv) NOT — ~
v) Right shift — >>
vi) Left shift — <<

Example :— i) AND operator

0 & 0 → 0
0 & 1 → 1
1 & 0 → 1
1 & 1 → 1

ii) OR operator iii) XOR operator iv) NOT

0 | 0 → 0
0 | 1 → 1
1 | 0 → 1
1 | 1 → 1

0 ^ 0 → 0
0 ^ 1 → 1
1 ^ 0 → 1
1 ^ 1 → 0

~0 → 1
~1 → 0

Q x = ~5 ; binary of 5 = 101

we store binary of 5 in 4 bytes :—

positive number hai 5 = 00000000 00000000 00000000 00000101

Now, we apply NOT operator on binary of 5 :—

5 = 11111111 11111111 11111111 11111010

negative number hai :—

NOTE :— Kisi bhi number ka binary karne par uska pahla bit

Subject: _____ Date: __/__/____

zero (0) ho to vo number positive hota hai. But agar pahala bit one (1) hai to vo number negative hai. So, negative number ko binary me karne ke liye hum 2's Complement ka use karte hai.

7. what is 2's Component.

⇒ 2's Component is a method to represent negative numbers in binary.

8. How we Convert a Number into 2's Component.

⇒ STEP 1:— Convert the number to binary.

Convert the positive value of the number into binary.

Example:—
Decimal 5 ⟶ Binary 0101 (used fixed bits, eg:, 4-bit)

STEP 2:— Find 1's Complement

Change all 0 ⟶ 1 and 1 ⟶ 0

Example:—
Binary 0101
1's Complement: 1010

Subject: Date: __/__/__

STEP 3:— Add 1 to the 1's Complement.

Add 1 to the result.

$$\begin{array}{r} 1010 \\ +\quad\ 1 \\ \hline 1011 \end{array}$$

Final Answer: 2's Complement = 1011

So, −5 in 4-bit 2's Complement = 1011

Solution of $x = -5$ is :—

STEP 1:— Convert the positive value of the number into binary.

Example :—

Decimal 5 :— 1010

STEP 2:— Find 1's

Solution of $x = -5$ is :—

$x =$ ~~000000~~

$x =$ 11111111 11111111 11111111 11111010

1's Complement :—

$x =$ 00000000 00000000 00000000 00000101

Subject: Date: / /

2's Complement :—

00000000 00000000 00000000 00000101

\+ 1

x= 00000000 00000000 00000000 00000110

x = -6 Ans

v) Right shift operator

Ex- x=53 >> 2;

53 = 00000000 00000000 00000000 00110101

After 2 time Right shift of 53 is :—

= 00000000 00000000 00000000 00001101

x = 13 Ans

vi) Left shift Operator

Ex:- x = 12 << 3

12 = 00000000 00000000 00000000 00001100

After 3 time Left shift of 12 is :—

= 00000000 00000000 00000000 01100000

x = 96 Ans