Operators

06 June 2023 23:03

Unary operators :- operators, which req. only 1 operand to fulfill their task are known as unary operators.

'+', '-', ++, --, sizeof(_) ✓

-- → decrement

++ → increment

```
int a = 10/+10;
int b = -10;
```

Arithmetic op :- requires 2 operands 2 assumes both operands are numbers(int, float, double)

'+', '-', '*', '/', '%' → remainder.

Brackets [()] → they are generally used for either accessing value, [] or increasing precedence () ↑

Assignment '=' used for assigning values.

prop → it returns value, it assigns.

Relational op.:- very similar to arithmetic op., because they also assumes that both operands are numbers.

only difference is they are used for comparison and they always return/evaluate to either 0 or 1 i.e. true or false.

Logical Operators :-

Logical AND → requires 2 operands.

→ similar to AND Gate

→ used for checking conditions.

$v_1 \&\& v_2$ → either eva. to 0 or 1
F or T

v_1	v_2	$v_1 \&\& v_2$ = res
0/F	0/F	0/F
1/T	0/F	0/F
0/F	1/T	0/F
1/T	1/T	1/T

If all values are true/1 then only it will give true.

It will stop executing as soon as

It will stop executing as soon as it gets first false value.

Logical OR '||' → similar to OR gate.

→ If any value 1 → 1.
else false/0.

v1	v2	res
0	0	0
0	1	1
1	0	1
1	1	1

} → T.T of '||'

Logical NOT. '!' → this works on single operant. (unary operator)

flip

T → F } T.T for '!'
F → T

In C++, any non zero value is true;
1, 0.1, -10, -4, 'A', true, !false,
are true values

false values are

are true values

In other words, only false values are 0 and false, rest all are true.

Increment / Decrement operators.

pre Increment

int ++a;

a=10;

pre Decrement

post Increment

int a=10;

a++;

post Decrement.

pre means → pehle increment/decrement hoga fir task hoga.

post means → pehle task hoga, fir increment/decrement hoga.

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

12 ~~11~~ ~~10~~
a

10
b

c
12