

## Assignment Operator

- Binary operator

- '='

- L value = R value  $\rightarrow$  exp/  
constant  
(or)  
variable

$\downarrow$   
cannot be expression  
cannot be constant  
must be a variable.

- Return value is assigned value.

# Arithmetic Operator

- unary

+, -

- binary

+, -, \*, /, %

- Modulus Operator

$a \% b \rightarrow$  gives remainder when  
a is divided by b

$\rightarrow$  Both operand must be  
of int type otherwise  
error

————→ sign of result is same  
as sign of first operand.

## Relational operator

.....

### Binary operator

$<$  —————→ less than

$<=$  —————→ less than (or) equal to

$>$  —————→ greater than.

$>=$  —————→ greater than (or) equal to

$!=$  —————→ not equal to

$==$  —————→ equal to

The result of every relational operator is either 0 (or) 1

# Logical Operators

.....

- i) Logical AND
  - ii) Logical OR
  - iii) Logical NOT
- binary
- unary

## Logical AND

.....

True → non zero value

False → zero

a	b	a & b
F	F	F
F	T	F
T	F	F
T	T	T

$a \& b = 1$ , when both operands are true, otherwise 0.

## Short circuiting in AND

---

0 &  $\square$

↳ if the first operand is 0, we don't need to evaluate 2<sup>nd</sup> operand.

## Logical OR

---

The o/p is 1, when at least one operand is non-zero

a	b	$a    b$
F	F	F
F	T	T
T	F	T
T	T	T

Shortcircuiting in OR

$1 || \square$

If the first operand is 1,  
we don't need to evaluate  
2<sup>nd</sup> operand.

## Logical Not (!)

---

!(true) = False

!5 = 0

!0 = 1

## Ternary operator

---

exp1 ? exp2 : exp3

exp1 is evaluated, if it is true

exp2 is evaluated otherwise

exp3 is evaluated.

Eg

$a = 12 > 10 ? 4 : 3$



$a = 3$



## Pre/post increment/decrement

---

Pre increment  $\longrightarrow ++a$

---

First increment then use

Post increment  $\longrightarrow a++$

---

First use then increment

Pre decrement  $\longrightarrow --a$

---

First decrement then use

Post decrement  $\longrightarrow a--$

---

First use then decrement