## Operator in C++



**Mohammad Tasin** 

# Agenda

- Arithmetic Instruction
- Classification of operators
- Unary operator
- Arithmetic operators
- Bitwise operators
- Relational operators
- Logical operators
- Assignment operators

### **Arithmetic Instruction**

- > Arithmetic Instruction :-
  - An instruction which is used to manipulate data using operators is known as arithmetic instruction



- 1. Unary operator
- 2. Binary operator
- 3. Ternary operator

## **Classification of operators**

- 1. Unary operator + ++ -- sizef()
- 2. Arithmetic operators \* / % +
- 4. Relational operators <> > <= >= !=
- 5. Logical operators && II !
- 6. Conditional operators ?:
- 7. Assignment operators = += -= \*= /= %=

[] → Subscript operator.

## **Unary operator**

Unary operator :- + - ++ -- sizef()

1. ++ (increment operator)

- 1. x++ (x=x+1) post increment operator
- 2. ++x(x=x+1) pre increment operator
- 2. -- (decrement operator)

- 1. x--(x=-x-1) Post decrement operator
- 2. --x(x=-x-1) Pre decrement operator
- Operand of Increment or Decrement operator must be a variable

## 3. sizeof()

- sizeof() is unary operator and keyword
- sizeof() operator return Data type, variable

and data of size

- 1. Data type
- 2. Variable
- 3. Constant/Data
- 1. Data type:
  - int x = sizeof(int);
- 2. Variable:int a;
  - int x = sizeof(a);

#### 3. Constant/Data:-

- int x = sizeof(10); = 4
- int x = sizeof(10.0); = 8
- int x = sizeof(10.0F); = 4
- int x = sizeof('A'); = 4

## **Arithmetic operators**

```
Arithmetic operators :- * / % - +
```

- int x = 10\*10;
  - printf("%d",x); // 100
- float n = 10/3;
  - printf("%f",n); // 3.000
- int z = 1249%10;
  - printf("%d",z); // 9
- 120.5%10 is Error

- int x = 10-10;
  - printf("%d",x); // 0
- int x = 10+10;
  - printf("%d",x); // 20
- ✓ Don't use real constant on modular (%) operator

## Bitwise operators

$$0 = 0 & 0$$

$$1 & 0 = 0$$

$$0 & 1 = 0$$

$$Ex := int x = 10 & 5$$

$$0 \mid 0 = 0$$

$$Ex := int x = 10 | 5$$

$$Ex := int x = 10 ^ 5$$

**Ex** :- int 
$$x = ~10$$

**Ex** :- int 
$$x = \sim -10$$

5. Right shift :- >>

$$Ex := int x = 20 >> 1$$

0000000 00000000 00000000 00010100



$$1010 = 10 \text{ Ans}$$

6. Left shift :- <<

$$Ex := int x = 20 << 1$$

0000000 00000000 0000000 00010100

<mark>0</mark>0000000 00000000 00000000 00010100<mark>0</mark>

101000 = 40 Ans

## Relational operators

Relational operators :- < > <= >= !=

Relational operators Result always 0 or 1

$$x = 10 < 20$$
; // 1  $\rightarrow$  Less then

 $x = 10 > 20$ ; // 0  $\rightarrow$  Grater then

 $x = 10 <= 20$ ; // 1  $\rightarrow$  Less then equal to

 $x = 10 >= 20$ ; // 0  $\rightarrow$  Grater then equal to

 $x = 10 != 20$ ; // 1  $\rightarrow$  Dose not Equal to

 $x = 10 == 20$ ; // 0  $\rightarrow$  Equal to

### Logical operators

Logical operators :- (&&)







1. AND :- **&&** 

Syntax:- Expression1 && Expression2

- True && True = True (1)
- True && False = False(0)
- False && True = False(0)
- False && False = False(0)

**Condition is False** 

2. NOT(unary) :-

- !False = True (1)
- !True = False(0)

int 
$$z = !10$$

$$z = 0$$

**Condition is False** 

3. OR :-

Syntax:- Expression1 || Expression2

- True | True = True (1)
- True | False = True (1)
- False | True = Ture (1)
- False | False = False(0)

Condition is True

## **Assignment operators**

Assignment operators :- = += -= \*= /= %=

4 = x; Error Invalid
5++; Error

Variable = anything———Valid

Ex:- x+=10; // x=x+10