Operators:

It is symbols used to perform Mathematical Operations. **8 type**

* **Arithmetic Operator:**
* To perform Addition, subtraction, multiplication etc.. ex: +,-,\*,/, %
* **Relational Operator:**
* SO relation operator is like to check whether it is less than, greater than, equal to
* Ex: <, >, <=, >=, ==, !=
* Return type will be Boolean.

(10>8) 🡺 true

* **Logical Operator**
* These operators will check the condition between two statements
* Return type will be Boolean.
* Ex: logical **AND** 🡺 &&, Logical **OR** 🡺||

(10>8) && (15<5) 🡺 true && false = false; (10>8) || (15<5) 🡺 true || false -> true

* **Bitwise Operator:**
* Bitwise always checks bott conditions whether first condition is true or false
* Ex: &, |
* (10>8) & (15<5) 🡺 true && false = false; (10>8) | (15<5) 🡺 true || false -> true
* **Assignment Operator:**
* It will assign values to right to the left side.
* Int I = 10;
* **Unary Operator**: i++. --i
* It is used to increase or decrease the data by one operand. Ex: ++ (increment operator) , -- (decrement operator).
* i++ 🡺 Post Increment
* ++i 🡺 Pre increment
* --i 🡺 pre decrement
* i-- 🡺 post decrement
* **Shift Operator** 10<<2 🡺 10\*2^2 = 40, 10<<3 🡺 10\*2^3 = 80
* Shift operator is like left shift and right shift
* Left shift << that means the value will be shift to left side of specified time
* Right >> that means value will be shift to right side

10>>2 🡺 10/2^2 , 10>>3 == 10/2^3

* **Ternary Operator**:
* These operators are used to check the conditions whether it is true or false.
* Ex: Condition? True: false

==

If(condition1 || condition2)

&& || 🡺 logical

& | 🡺 bitwise

= , ==

A = 5;

B = 10;

C = 6;

1. a<b 🡺 true
2. c > a 🡺 true
3. b < c 🡺 false

&& AND

(b<c) && (a<b) && (c>a) 🡺 false 🡺 logical AND

(b<c) & (a<b) & (c>a) 🡺 false 🡺 Bitwise AND

(b<c) || (a<b) || (c>a) 🡺 true 🡺 logical OR

(b<c) | (a<b) | (c>a) 🡺 true 🡺 Bitwise OR