★ OPERATORS

* operators are tokens used to perform different operations.

* operands are the variables or constant involved in an operation.

Eg :- a+b

* a and b are operands where as + is the operator

* Based on the number of operands required for the operation operators are classified into 3.

1) Unary operators.

* unary operator operates on a single operand.

* Unary +(positive) and unary -(negative) are considered as unary operators.

* This is used to represent sign of a number.

2) Binary operators

* Binary operators operate on two operands.

Eg :- arithmetic operators, relational operators etc.

3) Ternary operators. important

* Ternary operator operates on 3 operands.

Eg :- conditional operators (? :)

Based on the operations operators are classified into.

1. Arithmetic operators :-

* +(add), -(subtract), *(multiplication), /(division), %(modulus).

* Modulus operator performs division operation and returns reminder as output. /(division) operator performs division operation and returns quotient as output.

2. Relational operators :-

* Relational operators are used for comparing numbers.
* Relational operators are binary operators.
* Operation using relational operator returns either true or false value.
* True is represented by 1 and false is represented by 0.
* Operators < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), = (equal to), != (not equal to).
* A == B returns true only when both the variables 'A' and 'B' has same value or else the comparison returns false value.
* A != B returns true when both the variables 'A' and 'B' have different values. when both 'A' and 'B' have same value the comparision returns false value.

3. Logical operators :-

* Logical operators are used to combine two or more comparison.
* Different logical operators includes.

1) Logical AND ($\overset{\text{double ampersent}}{\&\&}$) operator

* When both the conditions returns true value the result will be true for all other condition it returns false value.

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	~~0~~ 1

2) Logical OR ($\overset{\text{pipe}}{||}$ operator)

* This operator returns true value when any of the condition either A or B returns a true value.
* This operator returns false value only when both A and B returns false value.

A	B	A||B
0	0	0
0	1	1
1	0	1
1	1	1

3) Logical NOT (!)

* This is a unary operation.
* This operator negate the result of the relational expression.

A	!A
0	1
1	0

4. INPUT / OUTPUT OPERATORS

* C ++ uses >> operator for input operation and this operator is called get from or extraction operator.
* << operator is used for output operation and this operator is also called as put to or insertion operator.

5. ASSIGNMENT OPERATOR (=) equal to

* This operator is used to store either a value to a variable or a value stored with in a variable to another variable.
  Eg: a = b , a = 5

6. ARITHMETIC ASSIGNMENT OPERATORS

* +=, -=, /=, *=, %= are considered as assignment operators.
* a=a+10 can be written as a+=10 or a=a/10 can be written as a/=10 or a=a-10 can be written as a -= 10.
* This is as also called as short hand form.

7. INCREMENT OPERATOR (++)

* This operator is used to increment a integer variable by 1 ;
* There are postfix increment and prefix increment.
* In postfix increment variable comes first following the operator. Eg: a++; a=a+1
* In prefix increment operator comes first followed by the variable. Eg: ++a ;

8. DECREMENT OPERATOR (--)

* This operator is used to decrement a integer variable by 1 ;
* There are postfix decrement and prefix decrement.
* In postfix decrement variable comes first following the operator. Eg: a--; a=a-1
* In prefix decrement operator comes first followed by the variable. Eg: --a ;

Important

```
#include<iostream.h>
void main()
{
int a=6;
cout<<a++<<++a;
cout<<a--<<--a;
}
```

OUTPUT

```
6886
```

★ CONDITIONAL OPERATOR (?:) important

* This is ternary operator works over 3 operands.
* condition will be checked first using the ?: operator.
* When the condition returns true the statement before the : will be executed. If the condition returns false the statement after the colon will be executed.

  Eg: r = m>50 ? 'p' : 'f'

* If the condition m>50 returns true value the variable 'r' will stored with the character 'p'. If the condition returns false the variable 'r' will be stored with the character 'f'.

Size of OPERATOR

* This operator can be used to check the number of bytes allocated for a variable or a data type or a constant.
* size of (int) returns 2 as result as the int data type allocates 2 bytes of memory. →perameter/argument
* size of (3.4) returns 4 as out put as the float data type will be allocated with 4 bytes of memory.

~~char a~~; ~~chara~~ char a;
~~size of (a)~~; size of (a);

* The above code returns 1 as output as the variable is declared in char data type and char data type allocates 1 byte of memory.