★ OPERATORS IN JAVA

Symbols that tell compiler to perform some operation.

Ex→ sum = a + b

Operand – On which operation is to be perform.

Operator – operation that is to be performed.

- Types of Operators
  1. Arithmetic Operators (Binary/Unary)
  2. Relational Operators
  3. Logical Operators
  4. Bitwise Operators
  5. Assignment Operators
  6. Ternary Operators.

- Arithmetic Operators

They are mathematical operations, these are of two types :-

1. Binary (atleast 2 operands)
   - `+` → 10 + 5 = 15
   - `-` → 10 - 5 = 5
   - `*` → 10 * 5 = 50
   - `/` → 10 / 5 = 2
   - `%` (Modulo) → 12 % 5 = 2

2. Unary (only 1 operand)
   - `++` → Increment
   - `--` → Decrement
   - * a = a+1 ——→ a++ or ++a
   - * a = a-1 ——→ a-- or --a

   Post Pre

* Pre-Increment (++a)
  ⇒ Value change → then use

* Post Increment (a++)
  ⇒ Value use → then change

Similarly, for Pre & Post Decrement also.

- **Relational Operators**

Operators that shows the relations between the operands.

1. == → equal → 10 == 10 → True
2. != → not equal to → 5 != 10 → True
3. > → ~~10 > 5~~ Greater → 10 > 5 → True
4. < → Smaller than → 5 < 10 → True
5. >= → Greater equal to → 10 <= 10 → True
6. <= → Smaller equal to → 5 <= 10 → True

- **Logical Operators**

Operators that check the logics of our programs.

1. Logical AND (&&) – अगर2 एक भी condition या relation False हुआ तो Final Ans = False.
2. Logical OR (||) – अगर2 एक भी condition या relation True है तो Final Ans = True होगा.
3. Logical NOT (!) – ! called as NOT operators.
   ! reverses the result.
   अगर2 True है relation तो Final Ans = False.
   अगर2 False है relation तो Final Ans = True.
   It can be applied on single statement.

- **Assignment Operators**

1. = → A = B → it assigns the value of B into A.
2. += → A = A + 10 → A += 10
3. -= → A = A - 10 → A -= 10
4. *= → A = A * 10 → A *= 10
5. /= → A = A / 10 → A /= 10

- Operator Precedence determines the order in which the operator in an expression are evaluated.
  * For eg - int x = 3*4-1;
  * In the above example, the value of x will be 11, not 9.
  * This happens becoz the precedence of * operator is higher than - operator.

S.No.	Operators	Precedence
1.	postfix increment & decrement	++ , --
2.	prefix increment & decrement & unary	++, --, +, -, ~, !
3.	multiplicative	*, /, %
4.	additive	+, -
5.	shift	<<, >>, >>>
6.	relational	<, >, <=, >=, instanceof
7.	equality	==, !=
8.	bitwise AND	&
9.	bitwise exclusive OR	^
10.	bitwise inclusive OR	|
11.	Logical AND	&&
12.	Logical OR	||
13.	Ternary	?:
14.	Assignment	<<=, >>=, >>>=, =, +=, *=, -=, /=, !=

- Associativity of Operators If an expression has two operators with similar precedence, the expression is evaluated according to its associativity.

S.No.	Operators	Precedence	Associativity
1.	post increment & decrement	++, --	Left to right
2.	pre Increment & decrement & unary	++, --, +, ~, -, !	right to Left
3.	multiplicative	*, /, %	Left to right
4.	additive	+, -	Left to right
5.	shift	<<, >>, >>>	Left to right
6.	relational	<, >, <=, >=, instanceof	Left to right
7.	equality	==, =!	Left to right
8.	Bitwise AND	&	Left to right
9.	Bitwise exclusive OR	^	Left to right
10.	Bitwise inclusive OR	|	Left to right
11.	Logical AND	&&	Left to right
12.	Logical OR	||	Left to right
13.	Ternary	?:	right to Left
14.	Assignment	=, +=, -=, *=, /=, %=, &=, ^=, |=, <<==, >>=, >>>=	right to Left

QUESTIONS

Ques 1.

```
int x = 2 , y = 5 ;
int exp1 = (x * y / x) ;
int exp2 = (x * (y / x)) ;
System.out.print (exp1) ;
System.out.print (exp2)
```

2 * 5 / 2 = 10 / 2 = 5

2 * (5 / 2) = 2 * 2 = 4

Ques 2.

```
int x = 200, y = 50, z = 100;
if (x > y && y > z) {
    System.out.print ("Hello");
}
if (z > y && z < x) {
    System.out.print ("Java");
}
if ((y + 200) < x && (y + 150) < z) {
    System.out.print ("Hello Java");
}
}
```

Java.

Ques 3.

```
int x, y, z;
x = y = z = 2;
x += y; // x = x + y = x = 2 + 2 = 4
y -= z; // x = y - z = 2 - 2 = 0
z /= (x + y); z = z / (x + y) = 2 / (2 + 2) = 2 / 4 = 0
System.out.print (x + " " + y + " " + z);
```

4, 0, 0

Ques 4. int x = 9, y = 12;
int a = 2, b = 4, c = 6;

int exp = 4/3 * (x + 34) + 9 * (a + b * c) + (3 + y * (2 + a)) / (a + b * y);

System.out.Print (exp);

4/3 * 43 + 9 * 26 + (3 + 12 * 4) / (2 + 48)
4/3 * 43 + 9 * 26 + 51 / 50
1 * 43 + 234 + 1
43 + 235
278

Ques 5. int x = 10, y = 5;
int exp1 = (y * (x/y + x/y));
int exp2 = (y * x/y + y * x/y);
System.out.print (exp1);
system.out.print (exp2);

y * (x/y + x/y)
5 * (10/5 + 10/5)
5 * (2 + 2)
5 * 4
20

y * x/y + y * x/y
5 * 10/5 + 5 * 10/5
50/5 + 50/5
10 + 10
20.

★ CONDITIONAL STATEMENTS

* It is type of statement which provides a condition to do work i.e, if condition is true then which work should be done & if false then which work should be done.

- **if-else Statement**

```
if (condition) {
    ____
    ____
}
else {
    ____
    ____
}
```

* एक statement में दो if statement check होती है after T.
* We can use multiple if's also.
* If there is single statement then there is no need for { }.
* But in companies it is necessary to use curly braces { } for single if statement.

- **else if Statements**

```
if (condition 1) {
    ____
    ____
}
else if (condition) {
    ____
    ____
}
else {
    ____
    ____
}
```

*In this statement, if a condition is true then it will check & execute the else if after T but F then check.

★ TERNARY OPERATOR

It is a type of operator which write if else statement in a single line or it combines the if else statement.

* Ternary means 3 operands.

Syntax → variable = condition ? statement 1 : statement 2 ;

False

True

: नहीं तो

★ SWITCH STATEMENT

```
Switch (variable) {
case 1:
    break;
case 2:
    break;
case 3:
    break;
default:
    break;
}
```

It can be changed. It depends what type of data we're using.

★ QUESTIONS

Ques 1. Write a Java program to get a number from the user & print whether it is positive or negative.

```
int num = sc.nextInt();
if (num > 0)
  {
    System.out.println("Number is Positive.");
  }
else { System.out.println("Number is Negative."); }
```

prints You have a fever if your temperature is above 100 & otherwise prints You don't have fever.

Ques 2. Write a Java program to get a number from the user & print whether it is positive or negative.

```
public class Solution{
  Public static void main (string args[]) {
    double temp = 103.5;
    if (100 < temp)
    {
      System.out.println ("You don't have fever");
    }
    else {
      System.out.println ("You have fever.");
    }
  }
}
```

Ques 3. Write a Java program to input week (1-7) & print day of week name using switch-case.

```
int week = sc.nextInt();
switch (week)
{
 case 1 : System.out.println ("Monday");
 break;
 case 2: System.out.println ("Tuesday");
 break;
 case 3: System.out.println ("Wednesday");
 break;
 case 4; System.out.println ("Thursday");
 break;
 case 5; System.out.println ("Friday");
 break;
```

```
case 6: System.out.println ("Saturday");
break;
cas 7: System.out.println ("Sunday");
break;
default : System.out.println ("Invalid Week No.");
}
```

Ques 4.
```
public class solution {
public static void main (string args[]) {
    int a=63, b= 36;
    boolean x= (a<b)? true : false;
    int y = (a>b)? a : b;
  }
}
```

false

63

Ques 5. Write a Java program that takes a leap year from user & print whether that year is leap year or not.

```
int year = sc.nextInt();
if (year % 4 == 0)
  {
    System.out.println ("Leap Year.");
  }
else
{
  System.out.println ("Not Leap Year");
}
```