Operators in Java

An operator is a symbol that specifies an operation to be performed.

1. **Arithmetic Operator**
2. **Relational Operator**
3. **Conditional Operator**
4. **Logical Operator**
5. **Assignment Operator**
6. **Increment/Decrement Operator**
7. **Bitwise Operator** etc.,

3.1 Arithmetic Operators

Java has five basic arithmetic operators. Each of these operators takes two operands.

Operator	Meaning	Example
+	**Addition**	**a+b**
-	**Subtraction**	**a-b**
*****	**Multiplication**	**a*b**
/	**Division**	**a/b**
%	**Modulo (Reminder Value)**	**a%b**

Operators as per precedence

Operator	Meaning
L-R	Left to Right
( )	For grouping the variables
***, /, %**	Multiplication, division and modulo
+ , -	Addition and subtraction

Type Casting

Converting from one specif datatype another specific datatype

Implicit Type Casting

Conversions done by java compiler automatically

OPERAND1	OPERAND2	RESULT
`int`	`int`	`int`
`int`	`float`	`float`
`float`	`float`	`float`
`float`	`double`	`double`
`double`	`float`	`double`
`int`	`double`	`double`
`double`	`double`	`double`

Explicit Type Casting

In a program conversion must be specify directly

Example `float x= (float) (int var1/int var2);`

Input of Java programs

Input of java program can be given in two ways

1. Outside of the program (Command Line arguments)
2. Inside of the program (Using Stream class)

1. Command Line Arguments

Input values can pass to java program from the command line. A command line argument is the information that directly follows the program's name on the command line when it is executed. They are stored as String array, passed to main method.

```
//program to calculate the expression a+b/c using command line arguments
class Argument
{
	public static void main(String args[])
	{
		int a,b,c;
		float f;
		a=Integer.parseInt(args[0]);
		b=Integer.parseInt(args[1]);
		c=Integer.parseInt(args[2]);
		f=(float)(a+b)/c; //type cast to float
		System.out.println("Expression a+b/c ="+f);
	}
}
```

The above program after compliation, the input as given as follows

OUTPUT

```
C:\>java Argument  5 2 2
Expression a+b/c = 3.5
```

2. Inside of the Program

Input can be given inside of the java program using `java.util.Scanner` class.

Program

```
//program to calculate the expression a+b/c using Scanner class methods

import java.util.Scanner;
class Scan
{
	public static void main(String arg[])
	{
	Scanner sc = new Scanner(System.in);
	System.out.print ("Enter the A value ");
	int a = sc.nextInt();
	System.out.print ("Enter the B value ");
```

```
        int b=sc.nextInt();
        System.out.print ("Enter the C value ");
        int c=sc.nextInt();

        float d=(float)(a+b)/c;
        System.out.println("Expression (a+b)/c  "+d);
        }
}
```

OUTPUT

```
Enter the A value 5
Enter the B value 2
Enter the C value 2
Expression (a+b)c is 3.5
```

3.2 Relational Operators

These operators are used to test for relationship, which help for decision making. All these expressions return a Boolean value either true or false.

Operator	Meaning	Example
==	Equality	a==b
!=	Not Equal	a!=b
<	Less than	a<b
>	Greater than	a>b
<=	Less than or equal to	a<=b
>=	Greater than or equal to	a>=b

3.3 Logical operators

These operators are used to evaluate more than one relational expression. Logical operators AND, OR and NOT.

Operator	Meaning	Example
&&	AND	a&&b
||	OR	a||b
!(unary)	NOT	!a

Example

A	B	A&&B	A||B	A ^B	!A
true	true	true	true	false	false
true	false	false	true	true	false
false	true	false	true	true	true
false	false	false	false	false	true

3.4 Conditional Operator - Ternay Operator

The Conditional Operator is otherwise known as **Ternary Operator** which is used to execute a true or false statement and is considered to be an alternative to the if..else construct.

Syntax `<test> ? <ture> : <false>`

where <test> is the condition to be tested. If the condition returns true, will execute true statement , otherwise false statement will be executed.

Example

```
big=a >  b ? a : b
```

Assignment Operator

Assignment operators used to assign a resultant value to a variable. A selection of assignment operators is given below

Syntax	Example
Variable = Constant	`a=10`
Variable1 = Variable2	`a=b`
Variable = Expression	`a=a+b`

Classification of Assignment Operators

Normal Assignment	Short Assignment
`x=x+y`	`x+=y`
`x=x-y`	`x-=y`
`x=x*y`	`x*=y`
`x=x/y`	`x/=y`

3.5 Increment and Decrement Operators - Unary Operators

These operators are used to Increment or Decrement a value by one. The increment and the decrement operatos can be prefixed or postfixed. The different types increment and decrement operators are given below.

PostIncrement Operator	Meaning
`a++`	First "a" value is Assign and then increment by one.

Example

```
a=10;
b=a++;
```

Result is : a=11
b=10

PreIncrement Operator	Meaning
`++a`	First Increment "a" by 1, then assign the value of "a" to b .

Example

```
a=10;
b=++a;
```

Result is : `a=11`
`b=11`

PostDecrement Operator	Meaning
`a--`	First "a" value is Assign and then decrement by one

Example

```
a=10;
b=a--;
```

Result is : **a=9**
b=10

PreDecrement Operator	Meaning
--a	First Decrement "a" by 1, then assign the value to "a" .

Example

```
a=10;
b=++a;
```

Result is : **a=9**
b=9

3.6 Bitwise Operators

The bitwise operators are inherited from C and C++. They are used to perform operations on individual bits in integers. A list of the bitwise operators are

Operators	Meaning
&	Bitwise AND
|	Bitwise Or
^	Bitwise XOR
<<	LeftShift
>>	RightShift
~ (unary)	Bitwise complement
x&=y	AND assignment
x|=y	OR assignment
x^=y	XOR Assignment

Program

```
class Bitwise
{
	public static void main(String arg[])
	{
		byte x=5;
		byte y=3;
		System.out.println("x & y is "+(x&y));
		System.out.println("x | y is "+(x|y));
		System.out.println("x ^ y is "+(x^y));
		System.out.println("x >> 2 is "+(x>>2));
		System.out.println("x << 2 is "+(x<<2));
	}
}
```

OUTPUT

```
x & y is 1
x | y is 7
x ^ y is 6
x >> 2 is 1
x << 2 is 20
```