

Lecture 07 – Java Operators

Java categorizes the operators in the following groups:

1. Arithmetic operators
2. Assignment operators
3. Relational operators
4. Logical operators
5. Bit-wise operators
6. Conditional Operators

1. Arithmetic Operators:

These operators are used with numeric values to perform common mathematical operations.

Table 1: Arithmetic operators

Operator	Meaning	Example
+	Add two operands	$x + y$
-	Subtract right operand from the left	$x - y$
*	Multiply two operands	$x * y$
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus – remainder of the division of left operand by the right	$x \% y$
++	Increment – Increases the value of a variable by 1	$x++$
--	Decrement – Decreases the value of a variable by 1	$x--$

2. Assignment Operators:

These operators are used to assign values to variables.

Table 2: Assignment operators

Operator	Example	Equivalent to
=	$x = 5$	$x = 5$
+=	$x += 5$	$x = x + 5$
-=	$x -= 5$	$x = x - 5$

<code>*=</code>	<code>x *= 5</code>	<code>x = x * 5</code>
<code>/=</code>	<code>x /= 5</code>	<code>x = x / 5</code>
<code>%=</code>	<code>x %= 5</code>	<code>x = x % 5</code>
<code>&=</code>	<code>x &= 5</code>	<code>x = x & 5</code>
<code> =</code>	<code>x = 5</code>	<code>x = x 5</code>
<code>^=</code>	<code>x ^= 5</code>	<code>x = x ^ 5</code>
<code>>>=</code>	<code>x >>= 5</code>	<code>x = x >> 5</code>
<code><<=</code>	<code>x <<= 5</code>	<code>x = x << 5</code>

3. Relational Operators:

These operators are used to compare two values.

Table 3: Relational operators

Operator	Meaning	Example
<code>></code>	Greater than – True if left operand is greater than the right	<code>x > y</code>
<code><</code>	Less than – True if left operand is less than the right	<code>x < y</code>
<code>==</code>	Equal to – True if both operands are equal	<code>x == y</code>
<code>!=</code>	Not equal to – True if operands are not equal	<code>x != y</code>
<code>>=</code>	Greater than or equal to – True if left operand is greater than or equal to the right	<code>x >= y</code>
<code><=</code>	Less than or equal to – True if left operand is less than or equal to the right	<code>x <= y</code>

4. Logical Operators:

These operators are used to combine conditional statements.

Table 4: Logical operators

Operator	Meaning	Example
<code>&&</code>	Logical AND — True if both the operands are true	<code>x && y</code>
<code> </code>	Logical OR — True if either of the operands is true	<code>x y</code>
<code>!</code>	Logical NOT — True if operand is false (complements the operand)	<code>!x</code>

5. Bit-wise Operators:

These operators are used to compare binary numbers.

For example, in the table below, let $x = 10$ (0000 1010 in binary) and $y = 4$ (0000 0100 in binary)

Table 5: Bit-wise operators

Operator	Meaning	Example
&	Bit-wise AND	$x \& y = 0$ (0000 0000)
	Bit-wise OR	$x y = 14$ (0000 1110)
~	Bit-wise NOT	$\sim x = -11$ (1111 0101)
^	Bit-wise XOR	$x \wedge y = 14$ (0000 1110)
>>	Bit-wise right shift	$x >> 2 = 2$ (0000 0010)
<<	Bit-wise left shift	$x << 2 = 40$ (0010 1000)

6. Conditional Operators:

The *conditional operator* is the only ternary operator (operator that takes three arguments) in Java. The operator evaluates the first argument and, if true, evaluates the second argument. If the first argument evaluates to false, then the third argument is evaluated. The conditional operator is the expression equivalent of the *if-else* statement.

Example 6

```
public class OperatorsDemo {  
  
    public static void main(String[] args) {  
        int a = 15, b = 12;  
        boolean c = a > b ? true : false;  
        System.out.println("c = " + c);  
    }  
}
```

Output:

```
c = true
```