

## Operators in Java

- Operators in Java are symbols that are used to perform operations on variables and values.

### Types:-

1. Unary Operator,
2. Arithmetic Operator,
3. Relational Operator,
4. Shift Operator,
5. Bitwise Operator,
6. Logical Operator,
7. Ternary Operator and
8. Assignment Operator

### Java Operator Precedence

Operator	Category	Precedence
Unary	postfix	<i>expr++ expr--</i>
	prefix	<i>++expr --expr</i>
Arithmetic	multiplicative	<i>* / %</i>
	additive	<i>+ -</i>
Shift	shift	<i>&lt;&lt; &gt;&gt; &gt;&gt;&gt;</i>
Relational	comparison	<i>&lt; &gt; &lt;= &gt;=</i>
	equality	<i>== !=</i>
Bitwise	AND	<i>&amp;</i>
	OR	<i> </i>
	XOR	<i>^</i>
Logical	AND	<i>&amp;&amp;</i>
	OR	<i>  </i>
Ternary	ternary	<i>? :</i>
Assignment	assignment	<i>= += -= *= /= %= &amp;= ^=  = &lt;&lt;= &gt;&gt;=</i>

## 1. Java Unary Operator

- The Java unary operators require only one operand.
- Unary operators are used to perform operations like
  - incrementing/decrementing a value by one
  - negating an expression

Operator	Meaning	Work
a++ a--	postfix	Print + operation
++a --a	prefix	Operation + print

## 2. Java Arithmetic Operators

- Java arithmetic operators are used to perform addition, subtraction, multiplication, and division.
- They act as basic mathematical operations.

Operator	Meaning	Work
+	Addition	To add two operands.
-	Subtraction	To subtract two operands.
*	Multiplication	To multiply two operands.
/	Division	To divide two operands.
%	Modulus	To get the area of the division of two operands.

## 3. Relational operators

- The **Java Relational operators** compare between operands and determine the relationship between them.
- The output of the relational operator is (true/false) boolean value

Operator	Meaning
==	Is equal to
!=	Is not equal to
>	Greater than
<	Less than
>=	Greater than or equal to

<=	Less than or equal to
----	-----------------------

#### 4. Shift Operator

- It is used to shift all bits in value to left/right side of specified number in times
- **Left Shift Operator**
  - The Java left shift operator << is used to shift all of the bits in a value to the left side of a specified number of times.
- **Java Right Shift Operator**
  - The Java right shift operator >> is used to move the value of the left operand to right by the number of bits specified by the right operand.

#### 5. Logical Operators

- These operators are used to perform logical "AND", "OR" and "NOT" operation.
- They are used to combine two or more conditions.

Operator	Meaning	Working
&&	AND	True --- All conditions need true False --- any one condition will be false
	OR	True --- any one condition is True False -- all condition false
!	Not	Invert conditions

#### 6. Bitwise Operators

Operator	Meaning	Work
&	AND Operator	True - All condition true False - Any one condition false
	OR Operator	True - any one condition is true False - all condition false
^	XOR Operator	False - Both condition same / T or F True - Have different values of conditions

## 7. Assignment Operators

- The **Java Assignment Operators** are used when you want to assign a value to the expression. The assignment operator denoted by the single equal sign `=`.
- In a Java assignment statement, any expression can be on the right side and the left side must be a variable name.

Operator	Example
<code>=</code>	<code>C = A + B</code> will assign value of <code>A + B</code> into <code>C</code>
<code>+=</code>	<code>C += A</code> is equivalent to <code>C = C + A</code>
<code>-=</code>	<code>C -= A</code> is equivalent to <code>C = C - A</code>
<code>*=</code>	<code>C *= A</code> is equivalent to <code>C = C * A</code>
<code>/=</code>	<code>C /= A</code> is equivalent to <code>C = C / A</code>
<code>%=</code>	<code>C %= A</code> is equivalent to <code>C = C % A</code>
<code>&lt;&lt;=</code>	<code>C &lt;&lt;= 2</code> is same as <code>C = C &lt;&lt; 2</code>
<code>&gt;&gt;=</code>	<code>C &gt;&gt;= 2</code> is same as <code>C = C &gt;&gt; 2</code>
<code>&amp;=</code>	<code>C &amp;= 2</code> is same as <code>C = C &amp; 2</code>
<code>^=</code>	<code>C ^= 2</code> is same as <code>C = C ^ 2</code>
<code> =</code>	<code>C  = 2</code> is same as <code>C = C   2</code>

## 8. Java Ternary Operator

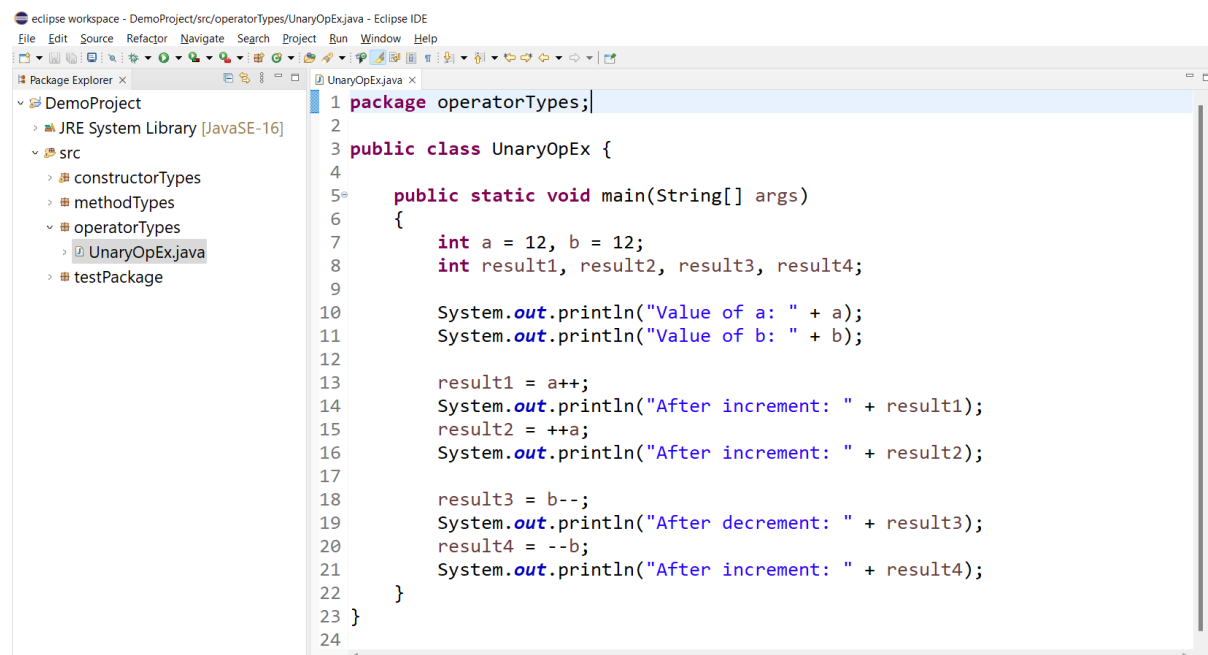
- The ternary operator (conditional operator) is shorthand for the if-then-else statement.

**variable = Expression ? expression1 : expression2**

Here's how it works.

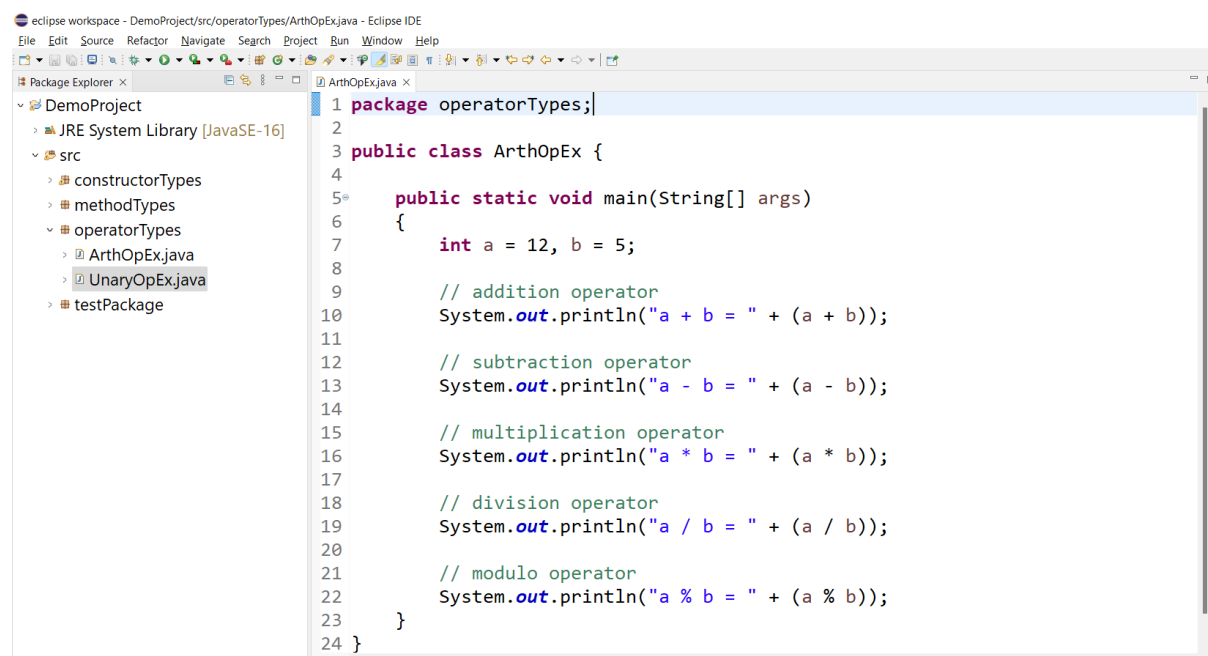
- If the Expression is true, expression1 is assigned to the variable.
- If the Expression is false, expression2 is assigned to the variable

## Unary Operator



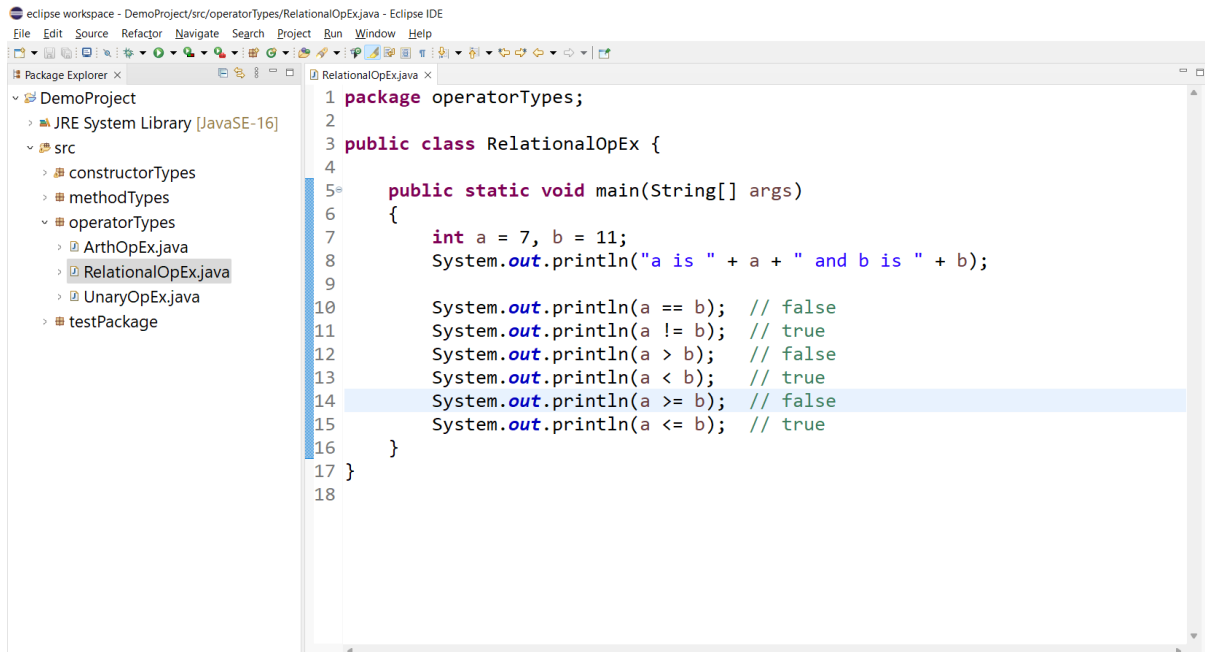
```
1 package operatorTypes;
2
3 public class UnaryOpEx {
4
5     public static void main(String[] args)
6     {
7         int a = 12, b = 12;
8         int result1, result2, result3, result4;
9
10        System.out.println("Value of a: " + a);
11        System.out.println("Value of b: " + b);
12
13        result1 = a++;
14        System.out.println("After increment: " + result1);
15        result2 = ++a;
16        System.out.println("After increment: " + result2);
17
18        result3 = b--;
19        System.out.println("After decrement: " + result3);
20        result4 = --b;
21        System.out.println("After increment: " + result4);
22    }
23 }
24
```

## Arithmetic Operator



```
1 package operatorTypes;
2
3 public class ArthOpEx {
4
5     public static void main(String[] args)
6     {
7         int a = 12, b = 5;
8
9         // addition operator
10        System.out.println("a + b = " + (a + b));
11
12        // subtraction operator
13        System.out.println("a - b = " + (a - b));
14
15        // multiplication operator
16        System.out.println("a * b = " + (a * b));
17
18        // division operator
19        System.out.println("a / b = " + (a / b));
20
21        // modulo operator
22        System.out.println("a % b = " + (a % b));
23    }
24 }
```

## Relational Operator



```
eclipse workspace - DemoProject/src/operatorTypes/RelationalOpEx.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x
DemoProject
  JRE System Library [JavaSE-16]
  src
    constructorTypes
    methodTypes
    operatorTypes
      ArthOpEx.java
      RelationalOpEx.java
      UnaryOpEx.java
    testPackage

RelationalOpEx.java x
1 package operatorTypes;
2
3 public class RelationalOpEx {
4
5     public static void main(String[] args)
6     {
7         int a = 7, b = 11;
8         System.out.println("a is " + a + " and b is " + b);
9
10        System.out.println(a == b); // false
11        System.out.println(a != b); // true
12        System.out.println(a > b); // false
13        System.out.println(a < b); // true
14        System.out.println(a >= b); // false
15        System.out.println(a <= b); // true
16    }
17 }
18
```

Logical Operator

Bitwise Operator

Shift Operator

Ternary Operator

Assignment Operator

