**Operators**Operators are the symbols that can be used to perform operations. Java operators are categorized into several types based on their function. Here’s an overview of the different categories **Explanation:**

* **Addition (+)**: Adds two numbers together.
* **Subtraction (-)**: Subtracts one number from another.
* **Multiplication (\*)**: Multiplies two numbers.
* **Division (/)**: Divides one number by another. In integer division, the result is an integer.
* **Modulus (%)**: Returns the remainder when one number is divided by another

public class ArithmeticOperatorsExample {

public static void main(String[] args) {

// Declaring two variables

int num1 = 10;

int num2 = 5;

// Addition

int sum = num1 + num2;

System.out.println("Addition: " + num1 + " + " + num2 + " = " + sum);

// Subtraction

int difference = num1 - num2;

System.out.println("Subtraction: " + num1 + " - " + num2 + " = " + difference);

// Multiplication

int product = num1 \* num2;

System.out.println("Multiplication: " + num1 + " \* " + num2 + " = " + product);

// Division

int quotient = num1 / num2;

System.out.println("Division: " + num1 + " / " + num2 + " = " + quotient);

// Modulus (Remainder)

int remainder = num1 % num2;

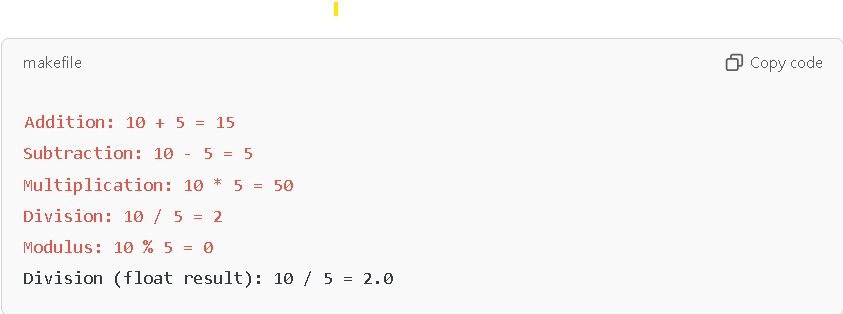
System.out.println("Modulus: " + num1 + " % " + num2 + " = " + remainder);

// Division with floating point result

int divisionResult = num1 / num2;

// Cast to double for accurate division

System.out.println("Division: " + num1 + " / " + num2 + " = " + divisionResult); } }

Output:

**Relational (Comparison) Operators in Java:**

Relational operators are used to compare two values or variables. They return a **boolean** value (true or false) based on the result of the comparison.

Here’s a list of relational operators in Java:

1. **== (Equal to)**: Returns true if both operands are equal.
2. **!= (Not equal to)**: Returns true if both operands are not equal.
3. **> (Greater than)**: Returns true if the left operand is greater than the right operand.
4. **< (Less than)**: Returns true if the left operand is less than the right operand.
5. **>= (Greater than or equal to)**: Returns true if the left operand is greater than or equal to the right operand.
6. **<= (Less than or equal to)**: Returns true if the left operand is less than or equal to the right operand.

**Example**:

public class RelationalOperatorsExample {

public static void main(String[] args) {

// Declaring variables

int a = 10;

int b = 5;

System.out.println( a== b);

System.out.println(a!=b);

System.out.println(a<b);

System.out.println(a<=);

System.out.println(a>b);

System.out.println(a>=b);

}}

Output:



**Logical Operators in Java**

Logical operators are used to perform logical operations on boolean values. They are often used in **conditional statements** and to combine multiple conditions.

Here are the three primary **logical operators** in Java:

1. **&& (Logical AND)**: Returns true if both conditions are true, otherwise returns false.
2. **|| (Logical OR)**: Returns true if at least one condition is true, otherwise returns false.
3. **! (Logical NOT)**: Reverses the boolean value. If the value is true, it becomes false, and if it is false, it becomes true.

**Syntax:**

* **&&**: condition1 && condition2
* **||**: condition1 || condition2
* **!**: !condition

public class RelationalOperatorsExample {

public static void main(String[] args) {

// Declaring variables

int a = 10;

int b = 5;

int c =20:

System.out.println(a>b && a>c);

System.out.println(a>b || a>c);

System.out.println(a>b != a>c);

}}

Output:

False

True

True

 **Left shift (<<)**: Multiplies the number by 2 power n

 **Right shift (>>)**: Divides the number by 2 power n and truncates the remainder.

class Main {

public static void main(String[] args) {

// Declaring variables

int a = 10;

int b = 5;

System.out.println(a>>b);

}}

Output:

0