Ministry of Higher Education
Jami University
Scientific assistantship
Faculty (Computer Science)
Department of Information Systems

Operators in java

Operators, Operands, and Operation in Java:

- **Operator**: An **operator** is a symbol that performs a specific operation on one or more **operands**. Examples of operators include `+`, `-`, `*`, `/`, `&&`, and `==`.

- **Operand**: An **operand** is the value or variable on which the operator operates. For example, in the expression `a + b`, `a` and `b` are the operands.

- **Operation**: An **operation** refers to the action performed by the operator on the operands. For instance, in `a + b`, the operation is **addition**.

Example:

```java
int a = 5;
int b = 10;
int sum = a + b;  // Here, '+' is the operator, 'a' and 'b' are operands, and addition is the operation.
```

- **In this example:**
  - **Operator**: `+` (performs addition)
  - **Operands**: `a` and `b` (values being added)
  - **Operation**: The addition of `a` and `b` (`5 + 10 = 15`)

Saifullah Haidari

In Java, **operators** are special symbols or keywords used to perform operations on variables and values. Java supports several types of operators, which can be grouped into categories based on the operation they perform.

Categories of Operators in Java:

1. **Arithmetic Operators**

2. **Relational (Comparison) Operators**

3. **Logical Operators**

5. **Assignment Operators**

1. **Arithmetic Operators**

These are used to perform basic mathematical operations.

Operator	Description	Example
`+`	Addition	`a + b`
`-`	Subtraction	`a - b`
`*`	Multiplication	`a * b`
`/`	Division	`a / b`
`%`	Modulus (Remainder)	`a % b`

Example:

```java
int a = 10;
int b = 5;
System.out.println(a + b);  // Output: 15
System.out.println(a % b);  // Output: 0
```

Saifullah Haidari

2. **Relational (Comparison) Operators**

These operators are used to compare two values.

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

Example:

```java
int x = 10;
int y = 20;
System.out.println(x > y);  // Output: false
System.out.println(x <= y); // Output: true
```

3. **Logical Operators**

These are used to perform logical operations, often combining multiple conditions.

Operator	Description	Example
`&&`	Logical AND	`a && b`
`||`	Logical OR	`a || b`
`!`	Logical NOT	`!a`

Example:

```java
boolean condition1 = true;
boolean condition2 = false;
System.out.println(condition1 && condition2);  // Output: false
System.out.println(condition1 || condition2);  // Output: true
System.out.println(!condition1);              // Output: false
```

4. **Bitwise Operators**

These perform bit-level operations on data.

Operator	Description	Example
`&`	Bitwise AND	`a & b`
`|`	Bitwise OR	`a	b`
`^`	Bitwise XOR	`a ^ b`
`~`	Bitwise NOT	`~a`
`<<`	Left shift	`a << 2`
`>>`	Right shift	`a >> 2`

Example:

```java
int num1 = 5;   // Binary: 0101
int num2 = 7;   // Binary: 0111
System.out.println(num1 & num2);  // Output: 5 (Binary AND)
System.out.println(num1 | num2);  // Output: 7 (Binary OR)
```

5. **Assignment Operators**

These operators are used to assign values to variables.

Operator	Description	Example
`=`	Assign	`a = 10`
`+=`	Add and assign	`a += 5`
`-=`	Subtract and assign	`a -= 3`
`*=`	Multiply and assign	`a *= 2`
`/=`	Divide and assign	`a /= 2`
`%=`	Modulus and assign	`a %= 2`

Example:

```java
int c = 10;

c += 5;  // c = c + 5

System.out.println(c);  // Output: 15
```

6. **Unary Operators**

These operators work with a single operand.

Operator	Description	Example
`+`	Unary plus (positive)	`+a`
`-`	Unary minus (negative)	`-a`
`++`	Increment (increase by 1)	`a++` or `++a`
`--`	Decrement (decrease by 1)	`a--` or `--a`

Example:

```java
int d = 5;
d++;  // d = d + 1
System.out.println(d);  // Output: 6
```

7. **Ternary Operator**

The ternary operator is a shorthand for `if-else` statements.

Operator	Description	Example
`? :`	Ternary conditional operator	`(condition) ? value1 : value2`

Example:

```java
int a = 10, b = 20;
int max = (a > b) ? a : b;
System.out.println(max);  // Output: 20
```

8. **Instanceof Operator**

This operator checks if an object is an instance of a specific class.

Operator	Description	Example
`instanceof`	Checks instance	`object instanceof Class`

Saifullah Haidari

Example:

```java

String str = "Hello";

System.out.println(str instanceof String);  // Output: true

```

15 Examples of Operators in Java:

1. **Addition (`+`)**:

   ```java

   int sum = 5 + 10;

   System.out.println(sum);  // Output: 15

   ```

2. **Subtraction (`-`)**:

   ```java

   int result = 20 - 8;

   System.out.println(result);  // Output: 12

   ```

3. **Multiplication (`*`)**:

   ```java

   int product = 7 * 6;

   System.out.println(product);  // Output: 42

   ```

4. **Division (`/`)**:

   ```java

   int division = 25 / 5;
   ```

   System.out.println(division); // Output: 5

5. **Modulus (`%`)**:
   ```java
   int remainder = 10 % 3;
   System.out.println(remainder);  // Output: 1
   ```

6. **Equality (`==`)**:
   ```java
   boolean isEqual = (5 == 5);
   System.out.println(isEqual);  // Output: true
   ```

7. **Not Equal (`!=`)**:
   ```java
   boolean notEqual = (5 != 10);
   System.out.println(notEqual);  // Output: true
   ```

8. **Logical AND (`&&`)**:
   ```java
   boolean result = (5 > 2) && (10 > 8);
   System.out.println(result);  // Output: true
   ```

9. **Logical OR (`||`)**:
   ```java
   ```

```
boolean result = (5 > 8) || (10 > 8);
System.out.println(result);  // Output: true
```

10. **Increment (`++`)**:

```java
int x = 5;
x++;
System.out.println(x);  // Output: 6
```

11. **Decrement (`--`)**:

```java
int y = 5;
y--;
System.out.println(y);  // Output: 4
```

12. **Bitwise AND (`&`)**:

```java
int bitwiseAnd = 5 & 3;  // 0101 & 0011 = 0001
System.out.println(bitwiseAnd);  // Output: 1
```

13. **Bitwise OR (`|`)**:

```java
int bitwiseOr = 5 | 3;  // 0101 | 0011 = 0111
System.out.println(bitwiseOr);  // Output: 7
```

14. **Left Shift (`<<`)**:

   ```java
   int leftShift = 5 << 1;  // 0101 << 1 = 1010
   System.out.println(leftShift);  // Output: 10
   ```

15. **Ternary Operator (`? :`)**:

   ```java
   int a = 10, b = 20;
   int max = (a > b) ?
   ```