



# Instructor Inputs

Session 2



## Session Overview

This session covers Chapter 2 of the book, Introduction to Java - SG. This session describes the various operators in Java. In addition, it explains the operator precedence in Java.

## Working with Operators

### Handling Tips

Start the session by discussing the scenario of Classic Jumble Word, where Sam needs to calculate the number of attempts in which the user gives a correct answer for a jumbled word. Then, ask the students what Sam should do to accomplish the desired task. Elicit the responses, and then inform them that for performing the calculations, Java provides various operators, such as arithmetic operators, assignment operators, comparison operators, logical operators, unary operators, bitwise operators, shift operators, and ternary operators.

Then, explain the students that operators can operate on one or more operands. Further, explain what are unary operators and binary operators. When you are discussing the topic, arithmetic operator, explain to the students that the arithmetic operator can be used with an integer or a floating point data type and discuss the information given in Additional Inputs topic.

In addition, explain the `instanceof` operator and bitwise OR operator with the help of the example given in the Additional Examples topic.

Further, while discussing the topic, bitwise operators, explain the following conversions to the students with the help of the information given in the Additional Inputs topic:

- Converting an integer number into a binary number
- Converting a binary number into an integer number

### Additional Inputs

Using the Arithmetic Operators

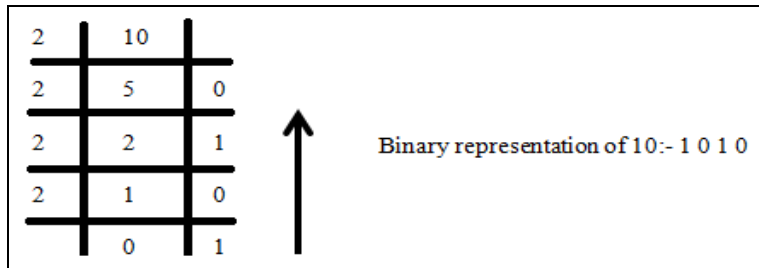
Consider the following code snippet:

```
int num1=1,num2=2;
float result=num1/num2;
System.out.println(result);
```

In the preceding code snippet, the output will be 0.0 and not 0.5. This is because `num1` and `num2` are integer values.

## Using the Bitwise Operators

The following figure demonstrates the conversion of an integer number into a binary number.



*The Conversion of an Integer Number into a Binary Number*

In the preceding figure, the integer number, 10, is divided by 2 to find out the equivalent binary number.

The following figure demonstrates the conversion of a binary number into an integer number.

Binary representation of 10:- 1 0 1 0

$$2^3 + 2^2 + 2^1 + 2^0 = 8 + 0 + 2 + 0 = 10$$

*The Conversion of a Binary Number into an Integer Number*

In the preceding figure, the binary number is 1010 and each 1 is multiplied by the equivalent power of 2 to find out the integer number.

## Additional Examples

### Using the Comparison Operators

The following code demonstrates the usage of the instanceof operator:

```
class Fruits{
int i,j;
}
class Flowers{
int m,n;
}
class TestInstanceOf{
public static void main(String args[]){
Fruits Mango = new Fruits();
Flowers Rose = new Flowers();
if(Mango instanceof Fruits) // instanceof operator
{
System.out.println("Mango is an instance of class Fruits.");
}
else{
System.out.println("Mango is not an instance of class Fruits.");
}
if(Rose instanceof Flowers){
```

```

System.out.println("Rose is an instance of class Flowers.");
}
else
{

System.out.println("Rose is not an instance of class Flowers.");
}
}
}

```

The output of the preceding code is:

```

Mango is an instance of class Fruits.
Rose is an instance of class Flowers.

```

## Using the Bitwise Operators

The following code demonstrates the usage of the Bitwise OR operator:

```

public class BitwiseDemo {
    public static void main(String args[]) {
        int a=10;
        int b=5;
        System.out.println(a|b);
    }
}

```

The output of the preceding code is:

```

3

```

## Activity 2.1: Working with Operators

### Handling Tips

Discuss the problem statement with the students.

The solution file, **Salarycalculator.java**, for this activity is provided at the following location in the TIRM CD:

■ Datafiles For Faculty\Activities\Chapter 02\Activity 2.1\Solution

## Using Operator Precedence

### Handling Tips

Ask the students to solve the following expression:

$$45+90/5*6$$

Use the whiteboard to list all responses from the students. Then, inform them that for performing such a calculation, Java implements operator precedence with the help of the information given in SG.

Thereafter, tell them that parentheses are used to override the order of precedence. Take one example, such as  $3*5+2$ , and tell them the output of the preceding expression is 17, but you want that the output should be 21, and then ask the students what needs to be done to accomplish the desired task. Elicit the responses, and then inform them that to accomplish the preceding task, they can use parenthesis, as shown in the following expression:

```
3*(5+2)
```

In addition, you can explain the order of precedence and implement precedence using parentheses with the help of the example given in the Additional Examples topic.

## Additional Examples

The following code demonstrates the usage of the order of precedence:

```
public class Test
{
    public static void main(String args[])
    {
        int a = 10;
        int b = 10;
        int c=9;
        float d = (a + ++b) * c / (a* b);
        System.out.println("The result is:" + d);
    }
}
```

The output of the preceding code is:

```
The result is:1.0
```

## FAQs

- *Can the == operator be used to determine whether two strings have the same value?*

Ans: No, the == operator cannot be used to determine whether two strings have the same value. You need to use the `equals()` method of the `String` class to compare two strings.

- *What is the difference between a unary plus operator and an arithmetic plus operator?*

Ans: A unary plus operator is used to indicate the positive value, whereas an arithmetic plus operator is used to add two numbers.

- *What is the difference between a logical AND (&&) operator and a bitwise AND (&) operator?*

Ans: When an expression involves a bitwise AND (&) operator, both the operands are evaluated. Whereas, when an expression involves a logical AND (&&) operator, then the first operand is evaluated; and if the first operand returns the value, `true`, then the second operand is evaluated; and if the second operand also returns the value, `true`, the logical AND returns the value, `true`. If the first operand returns `false`, then the second operand is not evaluated.