Java Relational Operators with Examples

Difficulty Level : Hard Last Updated : 02 Dec, 2021

Operators constitute the basic building block to any programming language. Java too provides many types of operators which can be used according to the need to perform various calculations and functions, be it logical, arithmetic, relational, etc. They are classified based on the functionality they provide.

Types of Operators:

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

Java Relational Operators are a bunch of binary operators used to check for relations between two operands, including equality, greater than, less than, etc. They return a boolean result after the comparison and are extensively used in looping statements as well as conditional if-else statements and so on. The general format of representing relational operator is:

Syntax:

```
variable1 relation_operator variable2
```

Let us look at each one of the relational operators in Java:

Operator 1: 'Equal to' operator (==)

This operator is used to check whether the two given operands are equal or not. The operator returns true if the operand at the left-hand side is equal to the right-hand side, else false.

Syntax:

```
var1 == var2
```

Illustration:

```
var1 = "GeeksforGeeks"
var2 = 20
var1 == var2 results in false
```

Example:

```
// Java Program to Illustrate equal to Operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 5, var2 = 10, var3 = 5;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 == var2: "
                           + (var1 == var2));

        // Comparing var1 and var3 and
        // printing corresponding boolean value
        System.out.println("var1 == var3: "
                           + (var1 == var3));
    }
}
```

Output

```
Var1 = 5
Var2 = 10
Var3 = 5
var1 == var2: false
var1 == var3: true
```

Operator 2: 'Not equal to' Operator(!=)

This operator is used to check whether the two given operands are equal or not. It functions opposite to that of the equal-to-operator. It returns true if the operand at the left-hand side is not equal to the right-hand side, else false.

Syntax:

```
var1 != var2
```

Illustration:

```
var1 = "GeeksforGeeks"
var2 = 20

var1 != var2 results in true
```

Example:

```
// Java Program to Illustrate No- equal-to Operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 5, var2 = 10, var3 = 5;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
```

```
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 == var2: "
                           + (var1 != var2));

        // Comparing var1 and var3 and
        // printing corresponding boolean value
        System.out.println("var1 == var3: "
                           + (var1 != var3));
    }
}
```

Output

```
Var1 = 5
Var2 = 10
Var3 = 5
var1 == var2: true
var1 == var3: false
```

Operator 3: ‘Greater than’ operator(>)

This checks whether the first operand is greater than the second operand or not. The operator returns true when the operand at the left-hand side is greater than the right-hand side.

Syntax:

```
var1 > var2
```

Illustration:

```
var1 = 30
var2 = 20

var1 > var2 results in true
```

Example:

```
// Java code to Illustrate Greater than operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 30, var2 = 20, var3 = 5;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 > var2: " + (var1 > var2));

        // Comparing var1 and var3 and
        // printing corresponding boolean value
        System.out.println("var3 > var1: "
                           + (var3 >= var1));
    }
}
```

Output

```
Var1 = 30
Var2 = 20
Var3 = 5
var1 > var2: true
var3 > var1: false
```

Operator 4: ‘Less than’ Operator(<)

This checks whether the first operand is less than the second operand or not. The operator returns true when the operand at the left-hand side is less than the right-hand

side. It functions opposite to that of the greater-than operator.

Syntax:

```
var1 < var2
```

Illustration:

```
var1 = 10
var2 = 20

var1 < var2 results in true
```

Example:

```
// Java code to Illustrate Less than Operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 10, var2 = 20, var3 = 5;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 < var2: " + (var1 < var2));

        // Comparing var2 and var3 and
        // printing corresponding boolean value
        System.out.println("var2 < var3: " + (var2 < var3));
    }
}
```

Output

```
Var1 = 10
Var2 = 20
Var3 = 5
var1 < var2: true
var2 < var3: false
```

Operator 5: Greater than or equal to (>=)

This checks whether the first operand is greater than or equal to the second operand or not. The operator returns true when the operand at the left-hand side is greater than or equal to the right-hand side.

Syntax:

```
var1 >= var2
```

Illustration:

```
var1 = 20
var2 = 20
var3 = 10

var1 >= var2 results in true
var2 >= var3 results in true
```

Example:

```
// Java Program to Illustrate Greater than or equal to
// Operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {
```

```
    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 20, var2 = 20, var3 = 10;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 >= var2: "
                           + (var1 >= var2));

        // Comparing var2 and var3 and
        // printing corresponding boolean value
        System.out.println("var2 >= var3: "
                           + (var3 >= var1));
    }
}
```

Output

```
Var1 = 20
Var2 = 20
Var3 = 10
var1 >= var2: true
var2 >= var3: false
```

Operator 6: Less than or equal to (<=)

This checks whether the first operand is less than or equal to the second operand or not. The operator returns true when the operand at the left-hand side is less than or equal to the right-hand side.

Syntax:

```
var1 <= var2
```

Illustration:

```
var1 = 10
var2 = 10
```

```
var3 = 9

var1 <= var2 results in true
var2 <= var3 results in false
```

Example:

```
// Java Program to Illustrate Less
// than or equal to operator

// Importing I/O classes
import java.io.*;

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Initializing variables
        int var1 = 10, var2 = 10, var3 = 9;

        // Displaying var1, var2, var3
        System.out.println("Var1 = " + var1);
        System.out.println("Var2 = " + var2);
        System.out.println("Var3 = " + var3);

        // Comparing var1 and var2 and
        // printing corresponding boolean value
        System.out.println("var1 <= var2: "
                           + (var1 <= var2));

        // Comparing var2 and var3 and
        // printing corresponding boolean value
        System.out.println("var2 <= var3: "
                           + (var2 <= var3));
    }
}
```

Output

```
Var1 = 10
Var2 = 10
Var3 = 9
var1 <= var2: true
var2 <= var3: false
```