

Operators

What is an operator?

- Operators are special symbols that perform specific operations on one, two, or three operands, and then return a result.
- Operators with one operand are called unary operators
- Operators with two operands are called binary operators
- Operators with three operands are called ternary operator

Operators available in java

- Assignment operators
- Arithmatic operators
- InstanceOf operator
- Relational operators
- Conditional operators
- Logical operators

Assignment operator

- The = is called the assignment operator.
- The = operator is used to assign the value present to its right to the operand present to its left.
- eg.,

```
int a=10, b;  
b=a;
```

Arithmatic operators

The basic arithmatic operators are

+	addition
-	subtraction
*	multiplication
/	division
%	modulo

String concatenation operator

- The + sign can also be used to concatenate two strings.
- Example
- String a ="hello", b="world";
- System.out.println(a + b) will result as **helloworld**

Increment or Decrement operators

- The `++` is the increment operator.
- The `--` is the decrement operator.
- There are two types based on the position of the operator with the operand.
- Postfix : `++` or `--` to the right of the operand
- Prefix : `++` or `--` to the left of the operand

Compound assignment operators

- The most commonly used compound assignment operators are
 - ✓ `+=`
 - ✓ `-=`
 - ✓ `*=`
 - ✓ `/=`

Relational operators

- The list of relational operators in java are

<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code><</code>	Less than
<code>></code>	Greater than
<code><=</code>	Less than or equal to
<code>>=</code>	Greater than or equal to

Conditinal operator

- The conditional operator is the ternary operator.
- The syntax of conditional operator is
`x=(boolean expression)?true:false ;`

InstanceOf operator

- The **instanceof** operator is used for object reference variables only.
- It can be used to check whether an object is of a particular type.
- The instanceof operator returns either true or false.
- Example:

```
Classname b = new Classname();
boolean a = b instanceof Classname;
```

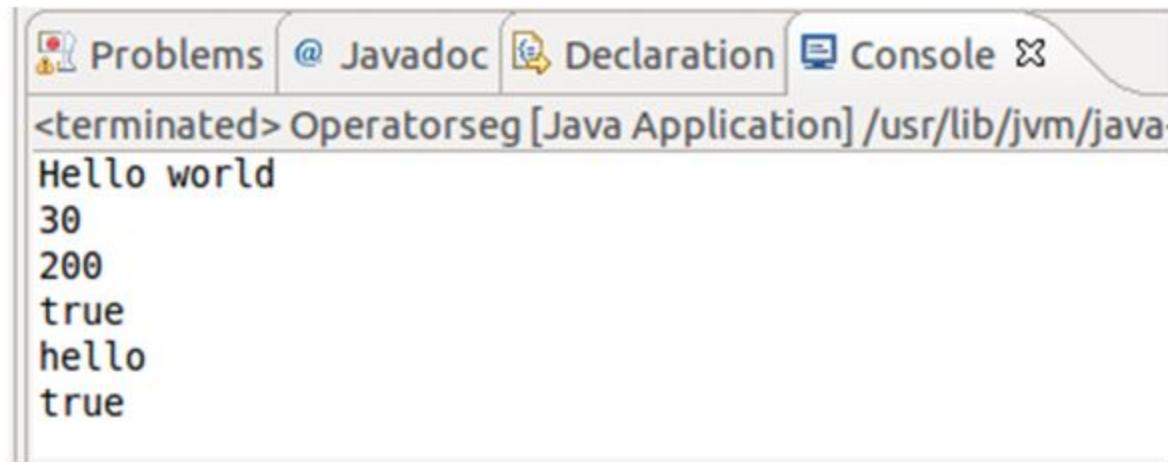
Logical operators

- Bitwise operators
 - &
 - |
 - ^
- Shorthand operators
 - && - logical AND
 - || - logical OR

Sample code for operators

```
package day7;
public class Operatorseg
{
    public static void main(String[] args)
    {
        int a,b,c;
        String str1,str2;
        //assignment operators
        a=10;b=20;c=0;
        str1="Hello";str2="world";
        //Arithmetic operations, concatenation operation,compound assignment
        System.out.println(str1+" "+str2);
        c=a+b;System.out.println(c);
        a*=b;System.out.println(a);
        //relational operators
        boolean x = a>b;System.out.println(x);
        //logical operators
        if(a>b&&b<c)
            System.out.println("hello");
        //conditional operator
        System.out.println(a>b?"true":"false");
    }
}
```

Sample output



The screenshot shows a software interface with a tab bar at the top containing 'Problems', '@ Javadoc', 'Declaration' (which is highlighted), and 'Console'. Below the tab bar, the title bar displays '<terminated> Operatorseg [Java Application] /usr/lib/jvm/java-'. The main area is a 'Console' window showing the following text output:

```
Hello world
30
200
true
hello
true
```

“

If you want to fly, give up everything
that weighs you down.

”

Buddha