COURSEWARE



# Operators

## Contents



## Overview

Java uses operators to perform tasks like, arithmetic calculations, assignment or a unary operation (an operation with only one operand, i.e. a single input). Below you can find a table of each operator and what function it serves.

| Operation Type | Operator | Function |
|---|---|---|
| Assignment | = | Assign value |
| Arithmetic | + | Plus |
| Arithmetic | - | Minus |
| Arithmetic | / | Divide |
| Arithmetic | * | Multiply |
| Arithmetic | % | Modulus |
| Unary | + | Indicates positive value |
| Unary | - | Indicates negative value |
| Unary | ++ | Increments by 1 |
| Unary | -- | Decrements by 1 |
| Unary | ! | Not operators |

These operators are very useful in Java programming, here we will be going over how to use each of them.

### Assign Value

```
int number = 7;
System.out.println(number);
```

The above code declares a variable with the reference name `number` of type `int` and assigns it the value of 7.
The code then prints the value stored in `number` to the console.

## Plus

```
int num1 = 3;
int num2 = 4;
int result = num1 + num2;
System.out.println(result);
```

The above code declares a variable with the reference name of `num1` and assigns it a value of 3, then declares a variable with the reference name of `num2` and assigns it a value of 4.
The code then declares a variable with the reference name `result` and assigns the value as `num1 + num2`, which in this case would be 7.
Finally the code prints the value stored in `result` to the console.

## Minus

```
int num1 = 5;
int num2 = 4;
int result = num1 - num2;
System.out.println(result);
```

The above code declares a variable with the reference name of `num1` and assigns it a value of 5, then declares a variable with the reference name of `num2` and assigns it a value of 4.
The code then declares a variable with the reference name `result` and assigns the value as `num1 - num2`, which in this case would be 1.
Finally the code prints the value stored in `result` to the console.

## Divide

```
int num1 = 12;
int num2 = 3;
int result = num1 / num2;
System.out.println(result);
```

The above code declares a variable with the reference name of `num1` and assigns it a value of 12, then declares a variable with the reference name of `num2` and assigns it a value of 3.
The code then declares a variable with the reference name `result` and assigns the value as `num1 / num2`, which in this case would be 4.
Finally the code prints the value stored in `result` to the console.

## Multiply

```
int num1 = 3;
int num2 = 4;
int result = num1 * num2;
System.out.println(result);
```

The above code declares a variable with the reference name of `num1` and assigns it a value of 3, then declares a variable with the reference name of `num2` and assigns it a value of 4.
The code then declares a variable with the reference name `result` and assigns the value as `num1 * num2`, which in this case would be 12.
Finally the code prints the value stored in `result` to the console.

## Modulus

Modulus divides one number by another and then returns the remainder.

```
int num1 = 12;
int num2 = 5;
int result = num1 % num2;
System.out.println(result);
```

The above code declares a variable with the reference name of `num1` and assigns it a value of 12, then declares a variable with the reference name of `num2` and assigns it a value of 5.
The code then declares a variable with the reference name `result` and assigns the value as `num1 % num2`, which in this case would be 2.
Finally the code prints the value stored in `result` to the console.

## Indicating a Positive Value

```
int num = 3;
num += num;
System.out.println(num);
```

The above code declares a variable with the reference name of `num` and assigns it a value of 3, then runs the `num += num` line.
This will make change the value of `num` to 6, and is effectively doing.

```
int num = 3;
num = num + num;
System.out.println(num);
```

After running the `num += num` line, the code prints out the value stored in `num` to the console.

## Indicating a Negative Value

```
int num = 3;
num -= num;
System.out.println(num);
```

The above code declares a variable with the reference name of `num` and assigns it a value of 3, then runs the `num += num` line.
This will make change the value of `num` to 0, and is effectively doing

```
int num = 3;
num = num - num;
System.out.println(num);
```

After running the `num -= num` line, the code prints out the value stored in `num` to the console.

## Incrementing by 1

```
int num = 5;
num++;
System.out.println(num);
```

The above code declares a variable with the reference name of `num` and assigns it the value of 5, the code then runs the ++ operation, incrementing the value of `num` by 1 which would change the value to 6.
The operation is effectively doing this:

```
int num = 5;
num = num + 1;
System.out.println(num);
```

After running the `num++` line, the code prints out the value stored in `num` to the console.

## Decrementing by 1

```
int num = 5;
num--;
System.out.println(num);
```

The above code declares a variable with the reference name of `num` and assigns it the value of 5, the code then runs the -- operation, decrementing the value of `num` by 1 which would change the value to 4.
The operation is effectively doing this:

```
int num = 5;
num = num - 1;
System.out.println(num);
```

After running the `num--` line, the code prints out the value stored in `num` to the console.

### Not Operation

```
boolean bool = true;
System.out.println(!bool);
```

The above code declares a variable with the reference name of `bool` and assigns it the value of true, the code then prints `bool` to the console whilst performing the not operation on it, which reverses it to false.
So the console ouput would be "false".

## Tutorial

There is no tutorial for this module.

## Exercises

### Calculator

1. Create a method that accepts two integers as input, then returns an integer that is the sum of these two parameters. (Addition)

2. Create the following additional three methods that each take two parameters:

    - Multiplication - which takes two numbers and returns the product.
    - Subtraction - which takes two numbers, then returns the result of the subtraction.
    - Division - which takes two numbers, then returns the result of the division.

3. Your division method may have returned the wrong result; This is called a `rounding error` and is quite common in most languages. this is because we were using ints rather than doubles. Modify the division method so that it takes Double parameters and then return a Double, if not already.

### Results

A person takes 3 exams in college, Physics, Chemistry, and Biology, each exam has a maximum of 150 marks. When all the exam marks have been added together, we can find the overall percentage that the person has got by multiplying their score by 100 and then dividing by 450.

1. Create the results class, this class has 5 variables, Physics, Chemistry, and Biology, total and percentage. This class should also have 2 methods:

    - Method 1 - displays the results that the person got for each exam and then the total mark. Try to make the output neat and bespoke for each exam.
    - Method 2 - which finds and displays the percentage that the person received for the exams overall.