Operators in Java

The Math Operators

Java provides a rich set of operators to use in manipulating variables. A value used on either side of an operator is called an **operand**.
For example, in the expression below, the numbers 6 and 3 are operands of the plus operator:

```
int x = 6 + 3;
```

Java arithmetic operators:
+ addition
- subtraction
*** multiplication**
/ division
% modulo

> Arithmetic operators are used in mathematical expressions in the same way that they are used in algebraic equations.

Addition

The + operator adds together two values, such as two constants, a constant and a variable, or a variable and a variable. Here are a few examples of addition:

```
int sum1 = 50 + 10;
int sum2 = sum1 + 66;
int sum3 = sum2 + sum2;
```

Subtraction

The - operator subtracts one value from another.

```
int sum1 = 1000 - 10;
int sum2 = sum1 - 5;
int sum3 = sum1 - sum2;
```

Multiplication

The * operator multiplies two values.

```
int sum1 = 1000 * 2;
int sum2 = sum1 * 10;
int sum3 = sum1 * sum2;
```

Division

The / operator divides one value by another.

```
int sum1 = 1000 / 5;
int sum2 = sum1 / 2;
int sum3 = sum1 / sum2;
```

In the example above, the result of the division equation will be a whole number, as **int** is used as the data type. You can use **double** to retrieve a value with a decimal point.

Modulo

The **modulo** (or remainder) math operation performs an integer division of one value by another, and returns the remainder of that division.
The operator for the modulo operation is the percentage (%) character.
Example:

```
int value = 23;
int res = value % 6; // res is 5
```

Try It Yourself

Dividing 23 by 6 returns a quotient of 3, with a remainder of 5. Thus, the value of 5 is assigned to the **res** variable.

Increment Operators

An **increment** or **decrement** operator provides a more convenient and compact way to increase or decrease the value of a variable by **one**.
For example, the statement **x=x+1;** can be simplified to **++x;**
Example:

```
int test = 5;
++test; // test is now 6
```

Try It Yourself

The **decrement** operator (--) is used to decrease the value of a variable by one.

```
int test = 5;
--test; // test is now 4
```

Try It Yourself

Prefix & Postfix

Two forms, **prefix** and **postfix**, may be used with both the increment and decrement operators. With prefix form, the operator appears before the operand, while in postfix form, the operator appears after the operand. Below is an explanation of how the two forms work:
Prefix: Increments the variable's value and uses the new value in the expression.
Example:

```
int x = 34;
int y = ++x; // y is 35
```

Try It Yourself

The value of x is first incremented to 35, and is then assigned to y, so the values of both x and y are now 35.
Postfix: The variable's value is first used in the expression and is then increased.
Example:

```
int x = 34;
int y = x++; // y is 34
```

Try It Yourself

x is first assigned to y, and is then incremented by one. Therefore, x becomes 35, while y is assigned the value of 34.

> The same applies to the **decrement** operator.

Assignment Operators

You are already familiar with the **assignment** operator (=), which assigns a value to a variable.

```
int value = 5;
```

This assigned the value 5 to a variable called **value** of type int.

Java provides a number of assignment operators to make it easier to write code.

Addition and assignment (+=):

```
int num1 = 4;
int num2 = 8;
num2 += num1; // num2 = num2 + num1;

// num2 is 12 and num1 is 4
```

Try It Yourself

Subtraction and assignment (-=):

```
int num1 = 4;
int num2 = 8;
num2 -= num1; // num2 = num2 - num1;

// num2 is 4 and num1 is 4
```

Try It Yourself

Similarly, Java supports multiplication and assignment (*=), division and assignment (/=), and remainder and assignment (%=).

The Math Operators

Java provides a rich set of operators to use in manipulating variables. A value used on either side of an operator is called an **operand**.
For example, in the expression below, the numbers 6 and 3 are operands of the plus operator:

```
int x = 6 + 3;
```

Java arithmetic operators:
+ addition
- subtraction
*** multiplication**
/ division
% modulo

> Arithmetic operators are used in mathematical expressions in the same way that they are used in algebraic equations.

Addition

The + operator adds together two values, such as two constants, a constant and a variable, or a variable and a variable. Here are a few examples of addition:

```
int sum1 = 50 + 10;
int sum2 = sum1 + 66;
int sum3 = sum2 + sum2;
```

Subtraction

The - operator subtracts one value from another.

```
int sum1 = 1000 - 10;
int sum2 = sum1 - 5;
int sum3 = sum1 - sum2;
```

Multiplication

The * operator multiplies two values.

```
int sum1 = 1000 * 2;
int sum2 = sum1 * 10;
int sum3 = sum1 * sum2;
```

Division

The / operator divides one value by another.

```
int sum1 = 1000 / 5;
int sum2 = sum1 / 2;
int sum3 = sum1 / sum2;
```

In the example above, the result of the division equation will be a whole number, as **int** is used as the data type. You can use **double** to retrieve a value with a decimal point.

Modulo

The **modulo** (or remainder) math operation performs an integer division of one value by another, and returns the remainder of that division.
The operator for the modulo operation is the percentage (%) character.
Example:

```
int value = 23;
int res = value % 6; // res is 5
```

Try It Yourself

Dividing 23 by 6 returns a quotient of 3, with a remainder of 5. Thus, the value of 5 is assigned to the **res** variable.