

## Arithmetic expressions

Arithmetic expressions in Java are composed with the usual operators `+`, `-`, `*`, `/` and the remainder operator `%`. Multiplication and division operations have higher priority than addition and subtraction. Operations with equal priority are performed from left to right. Parenthesis are used to control the order of operation execution.

$$a + b / c \equiv a + ( b / c )$$
$$a * b - c \equiv ( a * b ) - c$$
$$a / b / c \equiv ( a / b ) / c$$

It is recommended to always use parenthesis to explicitly define the order of operations, so you do not have to remember which operation has higher priority.

A thing worth mentioning is *integer division*.

The result of division in Java depends on the types of the operands. If both operands are integer, the result will be integer as well. The unintended use of integer division may therefore lead to significant precision loss. To let Java perform real division (and get a real number as a result) at least one of the operands must be of real type.

For example:

$$3 / 2 \equiv 1$$
$$2 / 3 \equiv 0$$

because this is integer division. However,

$$3 / 2. \equiv 1.5$$
$$2.0 / 3 \equiv 0.66666666...$$

because 2. and 2.0 are real numbers. If `k` and `n` are variables of type `int`, `k/n` is integer division. To perform a real division over two integer variables or expressions you should force Java to treat at least one of them as real. This is done by *type casting*: you need to write the name of the type you are converting to before the variable in parentheses, for example `(double) k/n` will be real division with result of type `double`.

Integer division is frequently used together with the *remainder operation* to obtain the row and column of the item from its sequential index. Suppose you have a collection of 600 items, say, seats in a theater, and want to arrange them in 20 rows, each row containing 30 seats. The expressions for the seat number in a row and the row would be:

Seat number: `index % 30` (remainder of division of index by 30: 0 - 29)

Row number: `index / 30` (integer division of index by 30: 0 - 19)

where index is between 0 and 599. For example, the seat with index 247 will be in the row 8 with seat number 7.

The power operation in Java does not have an operand (if you write `a^b` this will mean bitwise OR and not power). To perform the power operation you need to call the `pow()` function:

`pow( a, b )`  $\equiv a^b$

Java supports several useful shortcuts for frequent arithmetic operations over numeric variables. They are:

`i++`  $\equiv$  `i = i+1` (increment i by 1)

`i--`  $\equiv$  `i = i-1` (decrement i by 1)

`a += 100.0`  $\equiv$  `a = a + 100.0` (increase a by 100.0)

`b -= 14`  $\equiv$  `b = b - 14` (decrease b by 14)

Note that, although these shortcuts can be used in expressions, their evaluation has effect, it changes the value of the operands.