 

**Java Lesson: Math Operations
Last Updated: 2/11/2022**[**mr Hanley**](http://hanley.co.nr)

**Objective:** The objective of this lesson is to expose the student to the mathematical operators and common Math methods used in java.

Many times when we are creating software we find the need to perform mathematical calculations.

//declare some variables to be used in the example below

int numTickets = 10, numApples = 12, numOranges = 10, total\_Items;

double costApple = .35, costOrange = .42, costTickets = 14, totalCost;

final int BASKETSIZE = 8; //final variables are called constants (ALL CAPS STANDARD)

double side1 = 10.2, side2 = 7.3, side3 = 12.5;

Java allows the following mathematical operators;

|  |  |  |
| --- | --- | --- |
| **Operator** | **Meaning** | **Example** |
| + | Addition | total\_Items =  |
| - | Subtraction | daysLeft = 365-daysPassed; |
| \* | Multiplication | totalCost = numTickets \*  |
| / | Integer Division: yields ONLY integer answers  | 7/3 = 25/BASKETSIZE =  |
| //How many baskets needed to store apples?int numBasketsApples = numApples/BASKETSIZE; 12/8 = 1//you try to figure out the # of orange baskets |
| / | Floating Point Division: yields fractional answers | 7.0/3.0 = 25.0/BASKETSIZE =  |
| % | Modulus or Remainder | 10 % 3 = 20 % 12 = int leftOverApples = numApples % BASKETSIZE; 12 % 8 = 421 % 8 =  |
| ^ | DOES NOT RAISE TO POWER | Use another command to raise to a power |

Java also provides a collection of math methods for your use.

What is a method? *A method is a sub program within a program. It can have 0-many inputs and 0-1 outputs or return values. Methods take control of the program until they finish executing. Sometimes methods* ***return*** *a piece of information*

?:Where are methods defined?

*Methods are defined inside the { } of someone’s .java file, also known as a class*

?:How do I pass information to a method?

*You pass information to a method by sending arguments or data elements to a methods parameters*

?:How do I receive information from a method? *Information comes back from a method in the form of a return type*

Ok, can I see an example?

Let’s call the Math.sqrt() method.

Looking up Math.sqrt() in the java api (application programmer interface) at java.sun.com

At web site <http://java.sun.com/javase/6/docs/api/index.html> I found the following

 Here is where the **parameter** is

static double [**sqrt**](http://java.sun.com/javase/6/docs/api/java/lang/Math.html#sqrt%28double%29)(double a)
          Returns the correctly rounded positive square root of a double value.

This is where the **return type** is

public class MyProgram1 {

 public static void main(String[] args) {

 System.out.println(“Here is the square root of 12”);

 double answer;

 answer = Math.sqrt(12); //”Pass” 12 to the square root method

 System.out.println(answer);

 //You try: find the square root of 29 using this technique

 }

}

Some of my other favorites;

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Return Type** | **Parameters** | **Example** |
| Math.pow | double | double base, double exponent  | Math.pow(5,3) = 53 |
| Math.max | double | double a, double b  | Math.max(3, 100) |
| Math.round | int | double a  | Math.round(1.6) = 2 |
| Math.floor | double | double a | Math.floor(1.75) = 1 |
| Math.ceil | double | double a | Math.ceil(6.1) = 7 |