There are a lot of nuances to writing methods in Java. On the surface it is a fairly simple concept. A method is basically a piece of code that performs a specific function in a modular manner. It can be invoked in a single line of code into an existing body of code without “gunking” up your existing body of code by making it less readable.

The terms “Pass by value” and “Pass by reference” have to do with how arguments are passed to a method. “Pass by value” means that whenever a variable is passed as an argument to a method it is a copy of that variable value and not technically the same variable. “Pass by reference” however is literally the same variable. I think of “pass by reference” like how G# and Ab are same key on a piano. The context around those notes might change which spelling is “correct” but in the end you hit the same button no matter what. It is important to note that Java is technically a “pass by value” language and does not offer true “pass by reference” functionality.

Parameter order is another important concept to understand when it comes to writing methods in Java. They always need to be referenced in the order that they were initialized in. I liken this to how string indexes work, where a number corresponds to the position of a character.

In the below example notice how “toppings” and “size” are always in the same order.

public class Paramexam {

    public static void main(String[] args){

        paramExam("Pepperoni", 16);

    }

    public static void paramExam(String topping, int size) {

    System.out.println("You ordered a " + topping + " pizza, " + size + "inch size");

    }

}