For this week I will be focusing on some of the finer points of methods in Java. My two topics of focus will be the advantages and disadvantages of “method overloading” as well as void vs non-void methods.

First off, “Method Overloading” in Java relates to a Java feature that will enable a class to allow a single method name to “house” multiple methods by taking them as different parameters.

I have included a simplified version of the code from this week’s video to demonstrate how it works.

public class Example\_002{

  public static void main(String [] args){

    Example\_002.printParam(52);

    Example\_002.printParam(52.00);

  }

  public static void printParam(int param\_01){

    System.out.println("int value = " + param\_01);

  }

  public static void printParam(double param\_01){

    System.out.println("double value = " + param\_01);

  }

As you can see the assignment of the data types in the parameters of the printParam( ) method is being used as two different methods. One advantage of overloading a method would be reducing cognitive load, and thus making it easier to read. Another advantage is in how it is “modular like” and can be easily reused for different scenarios.

As for Voids vs Non-Voids, the main difference is whether or not a method will return a value or just perform an action and not return any data. A void, for example, just performs an action and does not return any data. This is useful for when you just want to run functions and don’t need any data manipulation.

A Non-Void is used for calculations or some kind of data retrieval needs to happen. The difference in syntax between these two is very subtle but I will demonstrate below.

Void

void methodName() {

    // action here

}

Non-Void

int methodName() {

    // math and things

    return intValue;

}

Notice the use of “return” in the non-void and the use of “void” in the, well…void.