**Generics Lab**

What I Learned:

From this lab, I learned how to take a pre-existing class and modify it into a generic class. I was initially given the class DataSet and tasked with using it to create the generic class DataSetGen. My first step in creating DataSetGen was to copy the code from DataSet.java into a new file, DataSetGen.java. For valid code, which requires the name of the file to be the same as the class it contains, I changed the class header in DataSetGen.java from “public class DataSet” to “public class DataSetGen”.The first step to make the class generic was to add a type parameter to the class header, “public class DataSetGen<T>”. It was not until later on that I realized this would not be all I needed to change to the class header. The purpose of this assignment was to have the DataSetGen load any instance of a class that implements the interface Measurable, so I had to figure out a way to restrict T to be a class that implemented Measurable. After several tries and some research, I found that the solution is to have T extend Measurable, “public class DataSetGen<T extends Measurable>”.

Constricting T allowed me to continue modifying the rest of the class, DataSetGen, to be generic. There were just three modifications I made, which all involved changing lines from using “Measurable” to “T”. After all this, I ran the main class found in DataSetTester and successfully got the expected output in the console (as shown in the attached screenshot).

What I Would’ve Done Differently:

If I could have done this lab differently, one thing I would have done is try to learn as much as I can about the type parameter before I started working on the lab. I had difficulty trying to figure out how I could restrict the type parameter of DataSetGen to classes that only implemented Measurable. I could have saved a lot more time if I understood the type parameter more clearly.

How I Will Apply Generics in the Future:

I look forward to discovering ways to use generics in Java classes when coding future projects. It is a concept that I have a lot to learn about, but I feel that I have gained a sufficient understanding. Previous to this week’s module, I have seen the type parameter used in code, but I have not understood why it was used or what it did. Now, I think that I will be able to understand more code that has been written in Java.

Console Screenshot:

