Generics Lab

What I Learned:

From this lab, I learned how to take a pre-existing class and modify it into a generic class. First, I copied the DataSet.java and put it into the newly created DataSetGen.java. I made sure to change the header to DataSetGen before getting started. The first step was to make the class generic was to add a type parameter. This is done by using T. The header was changed to DataSetGen<T>. I had gotten the first half correct and just needed to do the next part. For the header, I then added extends Measurable which allows for DataSetGen to load any instance of a class that implements the interface Measurable. Constricting T allowed to to modify the DataSetGen class to be generic. The last steps for this lab were changing the lines of code from Measurable to using T instead. I then ran the tester and got the expected output.

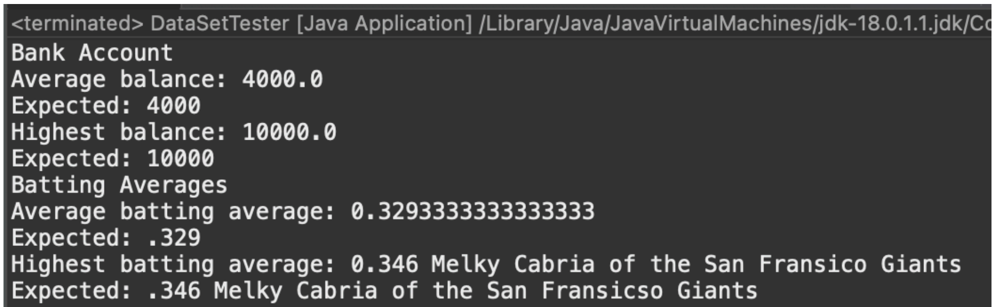
What I Would've Done Differently:

I need to slow down and actually try to understand what I'm doing. I found myself speeding through this lab without comprehending what I'm actually doing. Once I actually slowed down and took the time to understand what the goal was, I was able to complete the assignment much easier.

How I Will Apply Generics in the Future:

I feel generics will be used for code reuse in the future. You are able to write a method/class/interface once and use it for any type we want. It can also be used for type safety which allow for errors to happen are compile time and not at run time.

Console Screenshot:



```
<terminated> DataSetTester [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Co
Bank Account
Average balance: 4000.0
Expected: 4000
Highest balance: 10000.0
Expected: 10000
Batting Averages
Average batting average: 0.3293333333333333
Expected: .329
Highest batting average: 0.346 Melky Cabria of the San Fransico Giants
Expected: .346 Melky Cabria of the San Fransicso Giants
```