Lab 7, CSC/CPE 203
Due: 12/04

This lab explores *streams* through the implementation of a program to read in a set of numbers (representing points - specifically an image composed primarily of noise). Through implementing various computational operations on these points, they are transformed into a more coherent image (composed of points).

Streams are a useful tool to process collections. In particular, with a collection of data with a sequence of commands/operations, streams allow for efficient processing (and can even leverage multicore parallelism, which is beyond the scope of this lab, but good to know about). You can read more about them here: https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/stream/package-summary.html
Also check the examples we did in class.

Objectives

- To implement an entire program on your own which reads and writes data from and to a file
- To implement *streams* in order to compute various operations, including filter, map and collect.

Step 1

Start from one of your past assignments for class Point and modify based on functionality you want to apply (the coordinates must be double). Create a class that reads he file name from command line argument. This class must have methods that read from file, process the data, and write to a file. The deawPoints.java is given use this class to print your image on screen.

This data represents a large number of points in space with an x, y and z value. Consider modifying your existing `Point` class to use with this program, it may require adding more than just a data member for Z (see tasks below). Your code should represent all the points in a collection that can be processed as a stream.

The input data is stored in a file named **"positions.txt"** and is in the following format:

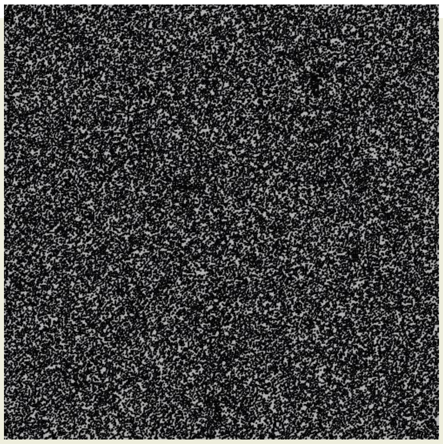
```
564.0, 414.0, 1
564.2765, 414.44946, 1
564.5011, 414.95673, 1
564.6649, 415.51572, 1
564.7596, 416.1191, 1
564.7776, 416.75833, 1
...
```

You can get a copy of these input points from Canvas. Write your code to take in the name of this input file as a **command line argument.** In IntelliJ you need to add file name in **Edit Configuration** from Run as argument.

To start with, just test reading in the points and writing them back out to a different file named **"drawMe.txt"**. Consider using `diff` on the two files to make sure that you can read in the data and write it out unchanged. You can also use the **helper program** provided, which displays points, using

Processing. The helper program can be found on Canvas and needs to be compiled with setting Processing appropriately to your lab (similar to lab7).

For your reference, to start with the point file should look like:



Step 2

After you have confirmed that you can read and write the points unchanged, work through the following operations *using streams*

- Remove all points that have a z value > 2.0.
- Scale down all the points by 0.5
- Translate all the points by {-150, -37}

Be sure that you use `collect, filter, map` of stream to complete these tasks. Now be sure to re-run the helper program to see the hidden image!

Submission

Demo and submit your files on the Canvas.

Rubric

1) Pass file name as argument	10 Points
2) Remove point > 2	10
3) Scale	10
4) Translate	10
5) Using Stream	20
6) Discover hidden image	20
7) Submission	30