CS 455 Lab 9: Enhance `DataAnalyzer` program

Fall 2020 [Bono]

Goals and Background

Note: this lab is due in your lab session for the week of 10/5 (8th week of classes). The goal of this lab is for you to practice more with exception handling. The lab involves enhancements to the `DataAnalyzer` example from Ch. 11 of the textbook and discussed in lecture.

For credit on these exercises, they must be written in the style the original program is written; for example, it uses exception handling, and the error-reporting and recovery happens in `main`.

Reading and reference material

- Ch. 11 of Horstmann text
- Section 11.5 of the text presents this example
- Exceptions and this example discussed in 10/7 lecture

Part 1 (1 checkoff pt)

The lab files are the DataAnalyzer example from the Exceptions chapter of the textbook, and some sample input files with and without errors.

Compile and run the program with the sample data files to see how it currently behaves.

Change the program so it still prompts for a new file name if the file is not found (i.e., it still does not exit on file not found), but now it will exit the program if it found any problems with the file format. (The original version prompts for a new file name in those cases.) It would, of course, report the error to the user before exiting, as it did before.

Part 2 (4 checkoff points total)

Change the program so it gives more descriptive error messages and distinguishes between more errors. Details given in the following paragraphs.

First, an error message that does not change as a result of your work: The following message will be the same as before, and happen in the same circumstances (except the program now exits on this error, because of the change you made for Part 1):

```
Bad data: Length expected
```

Instead of the other error messages the current program produces, the following messages will now be produced when these errors occur
Note: each one will be worth one point.

```
Bad data: Too many data values given
```

`Bad data: Too few data values given`

`Bad data: Non-floating point value given: 0-05`

> Shows an example of the output when the file contained `0-05` somewhere instead of a valid floating point number -- i.e., your message should include the invalid input detected.

`File not found: foo`

> Shows an example of the output when the file name entered was `foo`, but that file does not exist -- i.e., your message should include the file name given by the the user.

Checkoff for DEN students

Make sure you put your name and NetID in all the files you submit. No README required for this lab.

When you click the `Submit` button, it will be looking for and compiling (for source code) the files
`DataAnalyzer.java`, **and** `DataSetReader.java`.