



Programming Assignment 8

In this assignment you are to write a program to solve the following problems. As with all assignments, remember the following submission steps:

- Make sure your code passes at least all the provided JUnit tests
- Create and test Javadoc code documentation
- Save, commit, and push all code changes
- Confirm the latest code is visible via the “Files” section of your repository website
- Confirm that the repository is private, and that the instructor has Developer access

Not that for these problems, some JUnit tests that will be used for grading have not been provided as a part of the starter code. It is your responsibility to thoroughly test your code.

Problem a (PA8a.java)

Write a program that reads a stream of integers from a file and writes only the positive numbers to a second file. The user should be prompted to enter the names of both the input file and output file in `main()`, and then `main()` should attempt to open both files (providing an error if there is an error during this process). The `main()` method should then call the `process()` method to read all the integers from the input file and write only the positive integers to the output file. The `process()` method takes as arguments a `Scanner` to read from the input and a `PrintWriter` to write to the output. You can assume that if you are able to successfully open the input file, then there will only be integers in it.

You have been supplied JUnit tests for the `process()` method, as well as the output for several example file contents.

Problem b (PA8b.java)

Write a program that reads a stream of integers from a file and prints to the screen the range of integers in the file (i.e. [lowest, highest]). You should first prompt the user to provide the file name. You should then read all the integers from the file, keeping track of the lowest and highest values seen in the entire file, and only print out the range of values after the entire file has been read.

Importantly, unlike the previous problem, you can make no assumptions about the contents of the file. If your program cannot read the file, opens an empty file, or encounters a non-integer while reading the file, your program should output that the file is invalid.

You have been supplied JUnit tests for several invalid files (e.g. non-existent, empty, non-integer) as well as files that contain only integer values.