Lab 13 – Reading from and Writing to a File

1) Develop an ***algorithm*** for a Java program that reads data from a file. The program will involve two data files, `SoccerStats.txt` and `soccerSummary.txt`. The first file will be the input file that contains data involving soccer statistics. It includes the name of the soccer team, the number of goals, and the number of shots per game. The program will read in this data, which is separated by commas, from the text file one row at a time using a while loop.

2) This program will use an object to encapsulate the data values associated with a team. Here are the specifications for the **`Team.java`** class:

   - You will have 3 instance variables: name:String, numGoals:int, numShots:double
   - You will have a default constructor that sets all string values to an empty String and all numeric values to zero
   - You will have a parameterized constructor that passes in a name, numGoals, and numShots to set to the instance variables
   - You will have getters and setters for each instance variable

3) While developing your algorithm, please note the loop should use the `in.hasNextLine()` to check to see if there is a line to read in. Once the line is read in, you can use the `String.split(",")` method to split the line into an array of Strings. Each element of the resulting array contains one of the data values of a team, so use the values to create a new Team object.

4) The program will then find the maximum and minimum number of goals scored by the teams along with the name of the team that scored those goals. It will also find the average of the shots per game. The program will then write the results to `soccerSummary.txt`. The file's data should look like this:

   Maximum goals Scored: Monaco 15
   Minimum goals Scored: Athletic Bilbao 1
   Average shots per game: 15.565

5) Once your algorithm is complete, implement the program.

6) Navigate to the **Labs** package in IntelliJ and create a new package named **Lab13**. In the **Lab13** package, create a Java class named **Lab13.java** which will contain both a main method and a method named **processFile**, which will have a return type of void and take two parameters of type File.

```
public static void processFile(File inputFile, File
                    outputFile)throws FileNotFoundException{

     //Your code here
}
```

7) The main method will read two filepaths as command line arguments, create File objects from those filepaths, then call **processFile()**, passing the created File objects as arguments.

   Note: The main method **must** handle any exceptions. Your call to **processFile()** must be wrapped in a try/catch block. **processFile()** will perform all data manipulation.

   Note about IntelliJ: Place your input file in the project folder, this is the folder that contains the src folder. Do not place it in the src. Java cannot find it here. Your soccerSummary.txt file will be created when the program runs.

8) Use the algorithm you developed to implement the **processFile()** method.

9) Use the `Lab13Test.java` JUnit test cases to test your program. Take a screenshot of your running JUnit tests showing the tests passing and your Lab13.java.

10) Upload `Team.java` and `Lab13.java` to Gradescope and ensure that all tests pass.

11) Upload the screenshot and the `Team.java` and `Lab13.java` files to the submission area in Canvas.