**CSCI 4125/5125**

**Data Models and Database Systems**

**Fall 2023**

**Course Project**

**Phase1: Java Setup & Review (8/14)**

**Due: Tuesday, 8/22 @ 11:59pm**

**Reading:** SilberschatzChapters 1

**Submission Guidelines:**

1. This assignment is worth 30 points for all students.

2. All answers in the form of images or screenshots must be readable. Any additional files must be clearly referenced and labeled.

3. It is your responsibility to make sure all files are readable and submitted on time.

4. If you work with a team member, make sure you state this at the top of your submission. Each group member is still required to submit all required work and files.

**Submission:**

- Submit your Java source file and an output file (.csv or .txt) to Canvas.

**Task:**

We will use Java throughout the course project. Your task for right now is to write a Java program that does a simple evaluation for the data types of values in a file. Your Java program will eventually be able to automatically generate SQL insert statements for all the data we will use rather than you having to repetitively write many SQL insert statements. This will make your life a lot easier!!!

1. Read in the attached text (or csv) file, phase1\_fall2023.txt, line-by-line.
2. For each line, do the following:
   1. Use the split() method (Ex. String[] values = line.split(","); ) to store the (string) values in an array. Then, use a for-loop to iterate over them.
   2. For each value in the array, determine if the value is an integer, float, or string. *Hint*: Integer.parseInt(*value*) is one way you can determine if a value can be converted to an integer. Also use try{} catch{}.   
      *Hint*: Float.parseFloat(*value*) is one way you can determine if a value can be converted to a float. Also use try{} catch{}.  
      *Hint*: If the value is not an integer or a float, assume it is a string.   
      *Hint:* Some values may contain whitespace, which will interfere with your parsing. Leading and trailing whitespace can be removed with the trim().
3. Write your results for the data types to a text (or csv) file. For example, if my input is the following:

Hello, 99.99, World

Hello, World, 100, 2

My output file should look like:

String, Float, String

String, String, Integer, Integer

Hint: You can combine string elements with the join() method (Ex. String.join(",", myStrings) joins all the strings in the list myStrings using a comma).

**Requirements:**

* Your code must compile and run without any errors.
* This program must work for an input file containing records with any number of values and lines. Don’t hardcode line numbers or field (or column) positions.
* The output file must have the same number of lines as the input file. Do not write each individual datatype to a line.
* Your program cannot be longer than 100 lines. It should be ~50 lines so if it is significantly longer, your code is inefficient and probably difficult to read/understand.

Below is some pseudocode to give you a better idea of what this program should look like. I would also recommend using methods for the for-loop code.

Declare a variable (e.g., of type FileWriter) to write results to a file

Declare a variable (e.g., of type Scanner) to read the csv file

For each line in the file:

Initialize an output list

Split line on commas (i.e., line.split(",")) to get an array of values

For each value:

Remove whitespace.

If it is an integer, add "Integer" to the output list

If it is a float, add "Float" to the output list

Else, assume it’s a string. Add "String" to the output list

Generate an output string using String.join() and your output list

Write the output string to the output file.

Submit both your Java source file and your output file (.csv or .txt) to Canvas.