**605.201 Module 10 Java I/O Assignment**

Introduction:

This assignment is to provide practice in using the Java I/O techniques discussed in the Module 10 video lectures and readings. Although the main focus of this assignment is Java I/O techniques, Java design and implementation techniques discussed in earlier modules should be incorporated in to this assignment.

Problem:

Supplied is a data file from the US Census which contains data from US school districts and reports statistics related to child poverty (http://www.census.gov/did/www/saipe/data/highlights/2013.html). It is desired to have a summary report which calculate basic statistics at the state level.

Desired Implementation:

Java 8 implementation to read the supplied text data and produce a report similar to the below:

There should be two separate “programs” (two separate .java files each with a main method), one to read the text data file and write a reformatted file to be read by the second program which will create the report to standard out. Note before the report is displayed, a single line with “File: “ then the path of the input file for the report is displayed.

The first program is to create a data file (not the report) which provides a pre-processed view of the data supplied to it either by striping off the unneeded fields or by stripping off the unneeded fields and summing the data by state code. The numbers should not be formatted and no additional records should be produced.

The second program should read the file produced by the first program and produce the report in the format shown in the above image. This program should format the numbers and produce the file path information and column headings.

The first program will have 3 run-time parameters, the data source file path, the destination file path, and the number of records in the data file (13486) . If the program does not use the last run-time parameter, it should still accept it.

The second program will have 2 run-time parameters, the input file path and the number of records. If the program does not use the last run-time parameter, it should still accept it.

Features and Restrictions:

This assignment is an individual effort. Collaboration with other students on design approaches, implementation techniques, etc. as well as using the course's Discussion Board and other course resources are encouraged but the design, implementation, and submitted files must be your own creation.

A good reference for the Java 8 API is at: http://docs.oracle.com/javase/8/docs/api/

The programs should use standard (SE) Java 8 code and compile without errors or warnings. It should also run without errors or warnings when given valid input.

The programs should provide reasonable parameter validation (correct number of parameters, reasonable values, etc.).

The programs should not use any Java collection (ArrayList, Map, Vector, etc.) except standard Java arrays. Collections are introduced in a later module.

The file produced by the first program should not be deleted after running the report program.

The program's code should be reasonable formatted and commented as demonstrated so far in the course and reflected in the course’s Coding Standards document in the Syllabus & Course Information section of Blackboard.

Resources:

File: SmallAreaIncomePovertyEstData.txt – contains the small area poverty data. It is a standard 8-bit readable text file.

File: SmallAreaIncomePovertyEstLayout.txt – contains information about the field layout of the SmallAreaIncomePovertyEstData.txt file.

Submitted Files:

Please submit the following files in a single compressed zip file (.zip) using the following naming convention: Assignment10\_<JHUID><section number>. An example is: Assignment10\_jdealjr182

• Source files needed to recreate both of your programs

• The compiled .class files for your programs and any needed custom classes

• The supplied text files and any other additional files needed to run your programs