ASSIGNMENT 1C

Assignment 1C tests your knowledge of Sets and Maps (**Chapter 21**).

Part 1. Design a driver class called ***YourNameAssignment1C*** (replace ***YourName*** with your actual name) with the following exact methods (exact names, spelling, caps, parameters, returned values, functionality) in this order (do not merge the steps):

1. Create an ordered map data structure called **YourNameMap** (replace **YourName** with your actual name) - any type of map introduced in Chapter 21 like Map, LinkedListMap, or TreeMap (use your knowledge of the chapter and problem at hand to decide which type is better for this particular problem).

2. Read the data from the attached **Assingment1CData.txt** file that has the following format (with data separated by *commas* and *tabs*):
   Area, State   Population

   and store the *State – TotalPopulation* data into **YourNameMap** with the *State* as the key and *TotalPopulation* (the total from all the geographical areas from that state) as the value. For example, when you read *"Abilene city, Texas   123420"*, you are going to add *Texas-123420* to the **YourNameMap** map , then later when you read *"Allen city, Texas105623"*, you are going to add *105623* to the current *Texas* value which is *123420*, and so on until at the end you get *Texas- 15581916*.
   You should read the file only once and use your **YourNameMap** to keep track of the state population for the geographical area you have read/processed (so far).

3. Print the values from **YourNameMap** in alphabetical order of the states in a table with first column for the state and second column for the total population. You should use `printf` columns and not tabs to build your table columns.

Part 2. Implement the class in NetBeans IDE and JAVA: Create a JAVA project called ***YourNameAssignment1C*** (same name as the driver class) and add your code to ***YourNameAssignmen1C*** project/driver class. Your program should be **user-friendly** (prompt the user for the input using a descriptive message) and be **well-documented**/**commented** (have comments for every line of code).

Part 3. Create the screenshot document for your code and output: Create a Microsoft Word document from the **YourName-Assignment1C.docx** template and call it ***YourName-Assignment1C*.docx** (replace ***YourName*** with your actual name) Add to the document your **complete screenshots** of the NetBeans editor window showing the **complete JAVA source code** for *YourNameAssignment1C* class and **complete screenshots** the **complete output**. If the entire class JAVA source code or the output does not fit in one screenshot or the screenshots cannot be easily read, create multiple screenshots and add multiple screenshots to the screenshot document. Please keep the screenshots in order (look at the line numbers). If your output is longer than a line and does not fit on one screen, Wrap Text in your output panel. See the instructions on how to use the template.

Part 4. Submit your work: Submit ***YourName-Assignment1C.docx*** on eCampus under the **Assignment 1C.** Do not archive the files (e.g. no RAR, ZIP, etc.) or submit other types/formats of files (e.g. no CLASS, PDF, JAVA, etc.). Please take time after submitting to review the file and check it is correct and resubmit it if there are any errors.