Program 6: Files (50 Points)

**Objectives:**

The focus of this assignment is understanding File input and output

**Program Background:**

Files are used for long term, persistent storage. Files can store things like a previous program state, user preferences, or data to be processed. Many useful programs utilize file I/O for some purpose, such as reading a data file for interpretation/calculations, storing results, or both. Being able to quickly process and analyze large amounts of data is necessary for many fields. In this program you will be analyzing real crime data by state as reported on the FBI’s website. Data for each state is stored in its own CSV (comma separated values) file. Some sample files will be provided to you.

**Program Description:**

The program will allow the user to enter the name of a State, and if the CSV file for that state exists in the directory, it will read from that file and provide some crime statistics about that state and store the information. If the given State has already been analyzed it will read the stored data instead of processing the CSV file again.

The following classes are required:

* CityCrimeData
* StateCrimeData
* StateCrimeStats
* CrimeDataDriver

**UML DIAGRAM FOR AND DISCUSSION FOR CityCrimeData**

|  |
| --- |
| CityCrimeData |
| + cityName : String  + population : int  + violentCrimeCount : int  + murderCount : int  + robberyCount : int  + assaultCount : int  + propertyCount : int  + burglaryCount : int  + larcentyCount : int  + vehicleTheftCount : int  + arsonCount : int |
| <<constructor>> CityCrimeData(name : String, pop : int, vio : int, mur : int, rob : int, aslt : int, pro : int, bur : int, lar : int, veh : int, ars : int) |

The crime data in each file is broken down by major city. This class will store the data for one city. This is a very simple class. Many times we will just want a data type that is a collection of data without any methods or restrictions, which is called a tuple. Notice unlike more robust classes, all of the data members are public and therefore can be accessed directly.

Methods:

CityCrimeData(name : String, pop : int, vio : int, mur : int, rob : int, aslt : int, pro : int, bur : int, lar : int, veh : int, ars : int) – Class constructor. Sets data members to passed values.

**UML DIAGRAM FOR AND DISCUSSION FOR StateCrimeData**

|  |
| --- |
| StateCrimeData |
| * stateName : String * crimeByCity : ArrayList<CityCrimeData> |
| <<constructor>> StateCrimeData()  + readCrimeData( crimeFile : String ) **throws** FileNotFoundException  + processCrimeData( ) : StateCrimeStats  - calculateCrimeRate(propertyCount : int, population : int) : double |

A state’s crime data is a collection of all of its cities crime data, in this case stored in an ArrayList. This class has the methods to read in a CSV file, as well as process the crime statistics we are interested in.

Methods:

StateCrimeData() – Class constructor. Creates an ArrayList of CityCrimeData. The constructor does not set the state name; that will be determined from the CSV file read.

readCrimeData( crimeFile : String ) – This method takes the name of a CSV file, reads the contents, and stores them within the class’ data members. The CSV files first have the name of the State listed, every line after that is a comma separated sequence of values in the following format:

CITY NAME,POPULATION,VIOLENT CRIMES,MURDER,ROBBERY,ASSAULT,PROPERTY CRIME,BURGLARY,LARCENTY, VEHICLE THEFT,ARSON

For example, the first few lines of the Nebraska file look like this:

NEBRASKA

Alliance,8510,20,0,1,11,136,57,76,3,1

Ashland,2502,2,0,0,2,17,2,15,0,0

Aurora,4397,1,0,0,0,31,17,12,2,0

You will open this as a text file and read it using a Scanner, State first, then each city line by line, until all is read in. When reading the city data, read the entire line into a single String, and you can once again use the StringTokenizer class to parse out the data. In the previous use of StringTokenizer you used white space as a delimiter, in this case you will need to set the comma as a delimiter. StringTokenizer has a constructor to do just that:

StringTokenizer(str : String, delim : String)

The constructor takes the String to parse as well as a String that is a collection of characters to tokenize upon.

The nextToken() method will work the same as before, but parse on commas. Remember, nextToken() returns a String, most of the data are integer values, use Integer’s parseInt() method appropriately.

For each city data that is read in, create an instance of CityCrimeData and add it to the ArrayList.

processCrimeData( ) – Method for processing the crime data. This will go through the CityCrimeData to provide some statistics:

Population – Total population of the state

Violent Crimes – Total number of violent crimes in the state

Non-Vehicle Theft Crimes – Total number of theft crimes in the state. Non-Vehicle theft crimes are burglary, robbery, and larceny

City with Highest Property Crime Rate – While processing the cities, determines which city has the highest property crime rate. This is not property crime count, but rate. A crime rate is found by taking the (crime count/population)\*100,000. Use these values to create and return an instance of StateCrimeStats.

calculateCrimeRate(propertyCount : int, population : int) – Private method for calculating crime rate as described above.

**UML DIAGRAM FOR AND DISCUSSION FOR StateCrimeStats**

|  |
| --- |
| StateCrimeStats implements Serializable |
| * stateName : String * population : int * violentCrimes : int * nonVehicletheftCrimes : int * highestPropertyCrimeCity : String * highestPropertyCrimeRate : double |
| <<constructor>> StateCrimeStats( sta : String, pop : int, vio : int, thef : int, propCity : String, propRate : double)  + toString() : String |
|  |

A Serializable class that holds the crime statistics for a State

Methods:

StateCrimeStats( sta : String, pop : int, vio : int, thef : int, propCity : String, propRate : double) – Class constructor. Sets data members to the passed values.

toString( ) : String – Returns a String with the data formatted in the follow way (uppercase denotes data member values to be output):

State: STATENAME

Population: POPULATION

Violent Crimes: VIOLENTCRIMES

Non-Vehicle Theft Crimes: NONVEHICLETHEFTCRIMES

Highest Property Crime Rate: HIGHESTPROPERTYCRIMECITY - HIGHESTPROPERTYCRIMERATE

For example, Nebraska would look like this:

State: NEBRASKA

Population: 1171794

Violent Crimes: 4177

Non-Vehicle Theft Crimes: 37696

Highest Property Crime Rate: Scottsbluff - 4781.829050

**DISCUSSION FOR CrimeDataDriver**

See expected output below discussion, if your output is different in any way you are subject to losing points.

The user will be asked to enter the name of a State (with no spaces, New York would be entered as, “NewYork”). If a Serialized file of that name exists, read the StateCrimeStats object from the file and output it. If it does not exist then create a new StateCrimeData instance and read from the State’s CSV file, output the crime statistics and write the StateCrimeStats object to a Serialized file. If the CSV file does not exist, an error message will be output. See example runs for details.

**Example Runs (User input underlined)**

For these examples the following CSV files are in the same directory (and are provided to you for testing):

California.csv

Nebraska.csv

NewYork.csv

**Example Run #1**

Enter state name: Nebraska

Processing new data...

State: NEBRASKA

Population: 1171794

Violent Crimes: 4177

Non-Vehicle Theft Crimes: 37696

Highest Property Crime Rate: Scottsbluff - 4781.829050

Nebraska's stats saved.

**Example Run #2**

After the previous run, there is now the file Nebraska.ser, while the output is mostly the same, notice the message before the statistics are output. In this case, since the statistical data already existed, it was loaded instead of reprocessing.

Enter state name: Nebraska

Loading existing data...

State: NEBRASKA

Population: 1171794

Violent Crimes: 4177

Non-Vehicle Theft Crimes: 37696

Highest Property Crime Rate: Scottsbluff - 4781.829050

**Example Run #3**

Enter state name: California

Processing new data...

State: CALIFORNIA

Population: 31696400

Violent Crimes: 124598

Non-Vehicle Theft Crimes: 778255

Highest Property Crime Rate: Industry - 466816.143498

California's stats saved.

**Example Run #4**

Enter state name: NewYork

Processing new data...

State: NEW YORK

Population: 13933096

Violent Crimes: 70155

Non-Vehicle Theft Crimes: 288702

Highest Property Crime Rate: Seneca Falls Town - 12362.404742

NewYork's stats saved.

**Example Run #5**

Enter state name: Idaho

Processing new data...

Error reading crime data

**Assignment Questions**

Provide the answers to these questions in your submission directory within a file called Assignment6Questions.

1. What is beneficial about our input crime data being in a CSV file?

2. What benefit do we get in storing the crime statistics in a Serialized file? (This is not asking the benefit of simply storing the file, but specifically as a Serialized file.)

3. Describe an existing or potential collection of data that could be similarly processed in your own field of study.