**PROJECT Bank record generations 100 points**

**Objective** To write a program that parses and processes bank data from a file.

***PROJECT DESCRIPTION***

Bank of IIT has gotten their hands on some interesting data which will allow for possible loans to various clients from various regions.

Accompanying the labs specs is a csv (comma separated value) file named

**bank-Detail.csv** which contains valuable raw data to allow the bank to process loans based on client details from the file.

You need to parse the data and print record data for future loan considerations.

***Project Details***

-Create an abstract class called **Client.java** to allow for three abstract methods the bank needs to process. Name your methods readData(), processData() and printData(). No arguments are needed for your methods.

-Create a **BankRecords.java** file which will utilize the Client asbtract methods and generate ultimately the client records from the **csv** file.

The client file has the following header information

id {string}

age {numeric}

sex {FEMALE,MALE}

region {INNER\_CITY,TOWN,RURAL,SUBURBAN}

income {numeric}

married {NO,YES}

children {0,1,2,3}

car {NO,YES}

save\_act {NO,YES}

current\_act {NO,YES}

mortgage {NO,YES}

pep {YES,NO}

Create instance fields for each header item above in your class.

Include getter and setters for each instance field.

-Include code definitions for each method *declared* in your Client class as follows

Your **readData()** method should read in all the record data from the csv file in your

path into an ArrayList.

Your **processData()** method should take all the record data from your ArrayList and add the data into each of your instance fields via your setters. Use an array of objects to store record data for each instance field.

Your **printData()** method should print the first 25 records for various fields to the console via your getters. Print records for the following fields:

ID, AGE, SEX, REGION, INCOME, and MORTGAGE.

-Include headings in your print detail. Printing record detail should be in a neat columnar style format.

Make sure to include a try catch block when reading any file! Include proper exception handling. Include your project’s entire source code/csv file into a zip file, a jar file of your program, as well as your console output into a doc file for credit.