CIS 162 Lab 12
Fun with ArrayLists and File I/O

Objectives
After completing this lab, you should be able to:
- *Read data from a text file, and load it into an ArrayList.*
- *Write code that iterates through an ArrayList.*

Step 1: Create a New BlueJ Project – Lab12

Step 2: Download the Files
- Download the "CustomerRecords.txt" file and save it in the folder that BlueJ created for this new project. There are 1,000 entries! You will not see it from within BlueJ but you can see it `from` within the Windows Explorer.
- Download the provided CustomerDatabase.java to your project.

Step 3: Create a class called Customer
Instance Variables:
- first name (String)
- last name (String)
- email (String)

Constructor
- `public Customer(String first, String last, String email)` - a constructor that initializes all of the instance variables to the values given by the input parameters.

Methods
- `public String getFirstName ()` – return first name
- `public String getLastName ()` – return last name
- `public String getEmail ()` – return email
- `public String toString( )` - return a String containing the first name, las tname and email address (e.g. John Doe: john.doe@gmail.com
- Be sure to test this class thoroughly before moving on to the next class. No need to show how you did the testing of the Customer class

Step 4: Complete the provided CustomerDatabase class.

Class Instance Variables
Declare a private instance variable with a meaningful name:
- a reference to an ArrayList of Customer (don't instantiate the ArrayList yet)

Constructor

A *constructor* is a special method with the same name as the class and generally initializes the fields to appropriate starting values.

- `public CustomerDatabase ( )` - a constructor that instantiates the `ArrayList` of `Customer`. Do not add any customer to the ArrayList yet.

Mutator Methods

A mutator method performs tasks that may modify class fields.

- `public void readCustomerData(String filename)` - open the provided file and read all customer data. Read data one element at a time. Repeatedly, instantiate a new `Customer` and add it into the ArrayList.

Accessor Methods

- `public ArrayList<Customer> getDB()` – return the complete ArrayList of Customers. This is a single line of code.
- `public int getNumberCustomers ()` - return the number of customers in the ArrayList of Customers. This is single line of code.
- `public Customer findCustomer(String firstName, String lastName)` - if found, return the Customer that matches the provided first name AND last name. If not found, return `null`. The search should not be case sensitive… Hint: use the `equalsIgnoreCase` method to do your string comparisons.
- `public ArrayList<Customer> findCustomersWithSameEmailDomain(String domain)` – returns all Customer records whose email contains a specific email domain. For example, if domain is @google, you should return an ArrayList of all the Customers that contain @google in the email. Hint: use the `contains` method of the String class. If there are no records found for the domain entered as input parameter, the ArrayList returned should have zero records.

Step 5: Putting It All Together!

Note: I am providing the main method for you.):

These are the steps of the given main method. Look at the code to understand what it is doing.

1. Search and printout the record for Jack King, or a not found message if there is not such record.
2. Search and printout the record for Bill Gates, or a not found message if there is not such record.
3. Search and printout all customers that have a google email account

Step 6: Download the provided `CustomerDatabaseJUnit.java` file to your project and run all the test cases

Step 7: Download the provided `CustomerGUI.java` file to your project and run the GUI. <u>You don't have to do any changes to this class. This class is given so you can test your CustomerDatabase class.</u>

What to turn in? Use Blackboard to turn in your lab

- Upload your `Customer.java` file
- Upload your `CustomerDatabase.java` file
- A screenshot with the results of passing all the test cases of the CustomerDatabaseJUnit class

Grading Criteria

This lab is worth a possible 10 points.