Names: Romelia Acosta, Jesus Munoz
Date: October 23, 2024
CS 3331 – Advanced Object-Oriented Programming – Fall 2024
Instructor: Dr. Bhanukiran Gurijala
Assignment: Project Part 1

This work was done individually/as a team and completely on my/our own. I/we did not share, reproduce, or alter any part of this assignment for any purpose. I/we did not share code, upload this assignment online in any form, or view/received/modified code written from anyone else. All deliverables were produced entirely on my/our own. This assignment is part of an academic course at The University of Texas at El Paso and a grade will be assigned for the work I/we produced.

1. **Program Explanation**
   We created the banking system named El Paso Miners Bank. It comprises multiple classes including abstract and non-abstract classes such as Customer, Account, Savings, Check, Credit, and TransactionLogger plus the main class "RunBank" to loop through the system and handle user input. The program also tracks the user's transactions and views the activity of each registration session.
   We began by creating a UML Use Case Diagram and a UML Class Diagram to organize the system's entities. We use object-oriented programming principles such as inheritance, polymorphism, and abstraction. Focusing on these components, we were able to break the problem into smaller pieces and then piece it all together as we coded.

2. **What did I learn?**

Through this assignment, we learned how to design and simulate a simple bank system. We practice structuring classes and objects. The biggest learning curve was keeping track of the various objects/methods and objects throughout the classes. Deciding where to create an object/method as well as ensuring that it is created before it is being called was a challenge. We overcame this by implementing testing and debugging code throughout the classes. You may see these debugging statements as comments.

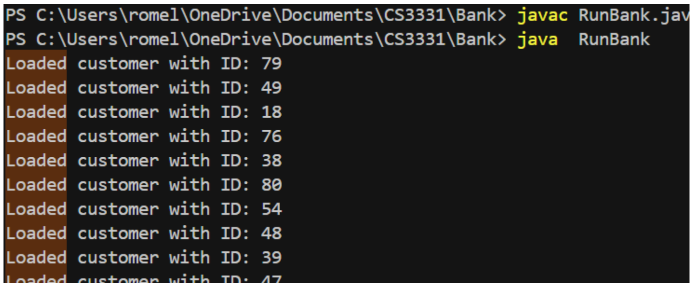
3. **Solution Design**
   What did you do in this program?
   In this program we organized a bank system into nine classes. We read a csv file to create persons, then customers, and use their information to create them their accounts. The program allows for transitions and bank information inquiry. We gae the program ability to collect all transaction information.

   We approached this by organizing in diagrams and updating them as we ran into new solutions during coding. We used objects, lists, and arrays.
4. **Testing**
   We used Black box testing to make sure that the interface works properly. We did this by looping through the system, inputting every possible "workable" input as well as inputs that would lead to a print statement stating that is an invalid input. We used White box testing by conducting unit

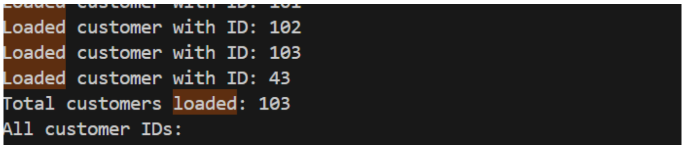
testing, for example: printing all of the account information imputed via csv scanner. This helped us improve our program to not have any gaps in our loop and account information.

5. **Test results**

Testing that every id is loaded correctly by printing as it is added. Test was successful:

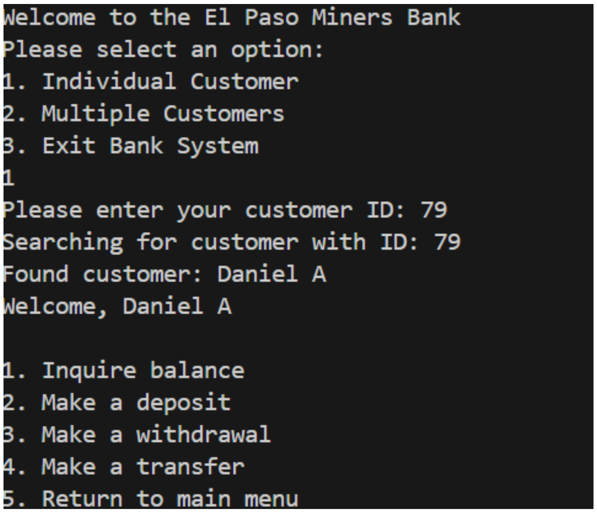


```
PS C:\Users\romel\OneDrive\Documents\CS3331\Bank> javac RunBank.jav
PS C:\Users\romel\OneDrive\Documents\CS3331\Bank> java  RunBank
Loaded customer with ID: 79
Loaded customer with ID: 49
Loaded customer with ID: 18
Loaded customer with ID: 76
Loaded customer with ID: 38
Loaded customer with ID: 80
Loaded customer with ID: 54
Loaded customer with ID: 48
Loaded customer with ID: 39
Loaded customer with ID: 47
```



```
Loaded customer with ID: 101
Loaded customer with ID: 102
Loaded customer with ID: 103
Loaded customer with ID: 43
Total customers loaded: 103
All customer IDs:
```

Testing that customer information input is accurate.



```
Welcome to the El Paso Miners Bank
Please select an option:
1. Individual Customer
2. Multiple Customers
3. Exit Bank System
1
Please enter your customer ID: 79
Searching for customer with ID: 79
Found customer: Daniel A
Welcome, Daniel A

1. Inquire balance
2. Make a deposit
3. Make a withdrawal
4. Make a transfer
5. Return to main menu
```

6. **Code Review**

We tested our code by looping through each choice in the RunBank class. As we looped, we coded test and debugging methods/print statements to solve malfunctions and errors.

(Note: Turn in all source code, output results (if applicable), reports, and all other required material specified by the assignment). Save this lab report at Team#_PA# – Do not turn in report with notes.)