Nadia Atiyeh
February 23, 2021
CS 3331 – Advanced Object-Oriented Programming – Spring 2021
Dr. Mejia
PA 3

I confirm that the work of this assignment is completely my own. By turning in this assignment, I declare that I did not receive unauthorized assistance. Moreover, all deliverables including, but not limited to the source code, lab report and output files were written and produced by me alone.

1. **Program Explanation**
   The purpose of this assignment was to upgrade the bank system that we had made in PA 2 and add more features to it like transactions completed from a reading file, adding a user directly from the console, reading from the file dynamically to populate my selected data structure to generating a bank statement for a specific user. To do this I mostly made changes to my RunBank class, and by adding another class called BankStatment that facilitated the generating of the text file. In this programming assignment I was also able to correct some of the issues I had from the last PA which was writing to the CSV with the updated balances from the program run. Tackling this PA I had to restructure most of my RunBank code and some of my other classes to handle the new way the Hashtable was being populated, which meant having to break up the new requirements by parts and then working with what I had and changing all of it to accommodate any changes that had to be made so that the system worked together as a whole. I split up the problem by first tackling the Hashtable populating problem, then moved on to adding users to the Hashtable from the console, then I tackled both reading from the transaction actions file, and writing to the bank statement files, and finally to writing to the CSV.

2. **What did I learn?**
   From this assignment I learned to read a file and populate a Hashtable without having to index through the file, with hardcoding, making my solution slightly more dynamic and accessible to different files. I think my solution can be improved when I read from the transaction actions file because I did not read from that file dynamically opting instead to hardcode my indexing, and I think with more time spent I can figure out how to do that now that I have experience with reading the bank users file in that way. Other ideas I have about solving this problem would be in my bank statement class, I want to find a way to format my statement so that when it writes to the file, fonts are formatted and look more like an actual document. To complete this lab I took about 2 weeks and a half, because I wanted to analyze my mistakes from the last PA, and fix them before even starting on the new requirements, I also took more time so that stress didn't play a factor on my code, and no incomplete work would be submitted.

3. **Solution Design**
   In this program like the last one I guide the users through a series of UI prompts and menus so that they can decide what they want to do at our bank of OOP, this time around the menus have more options and are built to handle errors slightly better than the last

time but not perfect yet as that is something, I really want to focus on in the next PA. My approach to solving this problem was similar to the other PA's by white box testing as I added code to my program to ensure everything being integrated correctly. Next, I split the problem up into sections to make it easier to connect everything in the end, starting with reading the bankUsers csv dynamically. To do this I iterated through the first line in the CSV to gather the column names, after that I iterated through what was read in the file matching each position in the column names to each position in what was read to populate my Hashtable, I encountered many problems when doing this however because the function array.split(","); gave me more read fields than column names which was something I had to maneuver around, but eventually after that solution was done I worked on changing and modifying the rest of the code to accommodate the dynamically read csv, and fixed the issue I was having with the Hashtable not returning the correct final balances as the end. In the end I ended up using two major data structures, the first being a Hashtable to fit all my users, and the other being an ArrayList to get all the transactions any user had made in a certain iteration to assist in the bank statement class when writing to the file. One assumption I did make in my program is that the transaction actions did not have any incorrect transactions in the file, that is something I plan on changing in the next PA, but for right now that is my only assumption.

4. **Testing**

   I tested my program through white-box testing because I would compile every 2 minutes to see if what I had written so far was working how I wanted it to, when doing my Junit testing, I also tested with white-box testing as I wrote my testcases knowing what I wanted my output to be. I think I did test my solution enough because I would go through every possible combination and see what the output was, my testing practices could have been improved by making a guide for myself, so that I didn't get lost on what step I was on when testing. The test cases I used would be to test reading the CSV file and then depositing, withdrawing, transferring, inquiring, and paying with random values, in my Junit tests I used mostly assertTrue(), or assertFalse(), but also a combination of assert with a specific value to see if my transaction methods had the correct return value when testing with edge cases, this time around I also tested with negative values to offset the program and ensure the correct output was given. Throughout my testing I broke my code many times trying to find a way to make what the user saw as user friendly as possible, and checking if all the associated class methods were working as they should, when breaking my program I saw that some of my transaction methods needed Override methods inside their specific classes which helped me see the code in a broader view.

5. **Partner Demo**

   Worked with: Diana Licon

   a. Some questions I had about my Partners functionality, was the decision to put certain commands in certain UI menus, another question would be how much of the logic is done in main vs. how much is done in the respective classes.
   b. One of the concerns I had about my partners functionality, was the programs' ability to handle exceptions like input mismatch exceptions in the main since the program does show a complier error method instead an error produced by the code written.

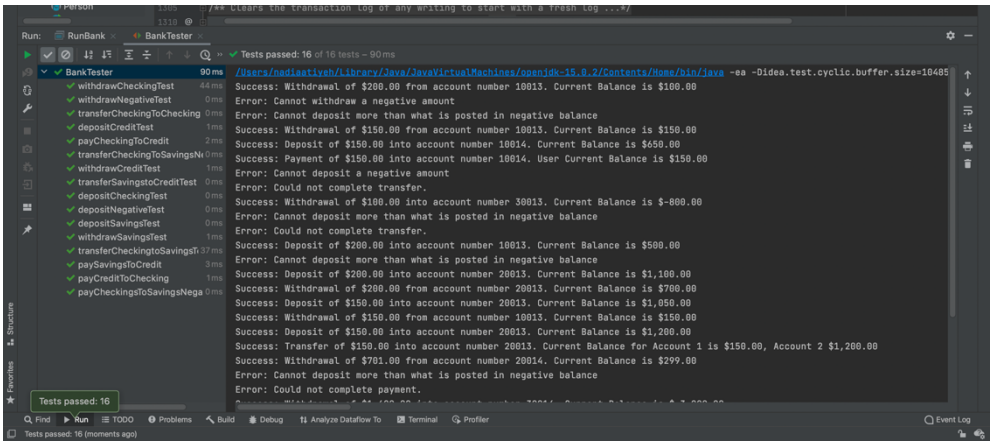
```
what account would you like to use
1)Checking
2)Savings
3)Credit
1
How much money would you like to send to Nadia Atiyeh?
e
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextDouble(Scanner.java:2413)
    at RunBank.main(RunBank.java:141)

Process finished with exit code 1
```

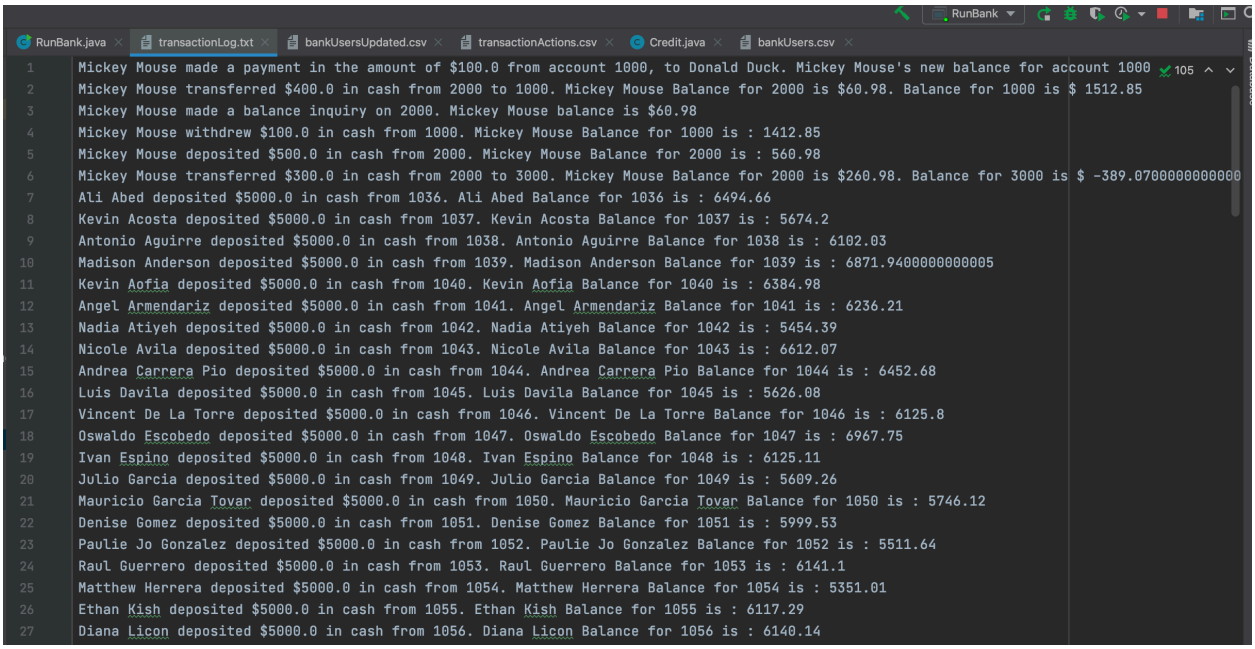
c. I tried to break this code by first going through the whole functionality of the program and testing it with correct values. Afterwards when everything was working as it could and we had taken note of things to change in the future, I tried to deliberately break the program, this included giving the wrong names, input mismatch exceptions, and checking if the loops were broken in the correct way.

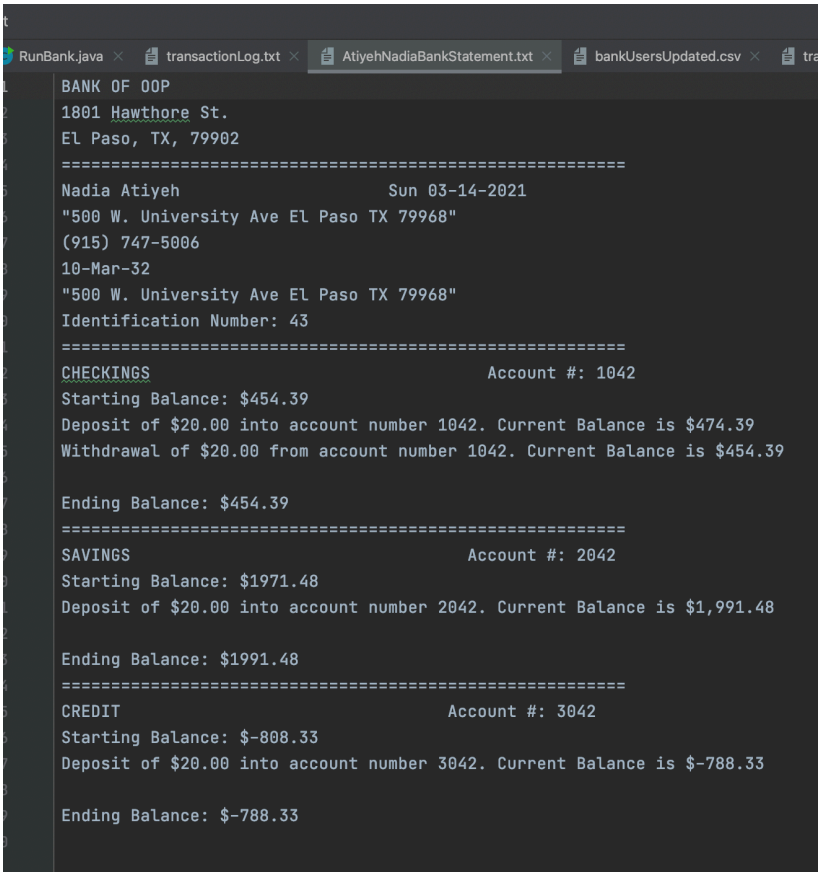
d. The way I tested this is a combination of both black box and white box testing in the sense that I know what the programmer meant when the code was made so I know how to break the program deliberately, but it was also black box testing in that I didn't know the order of the UI and I was not accustomed to how the commands and prompts were written and read.

6. Test results

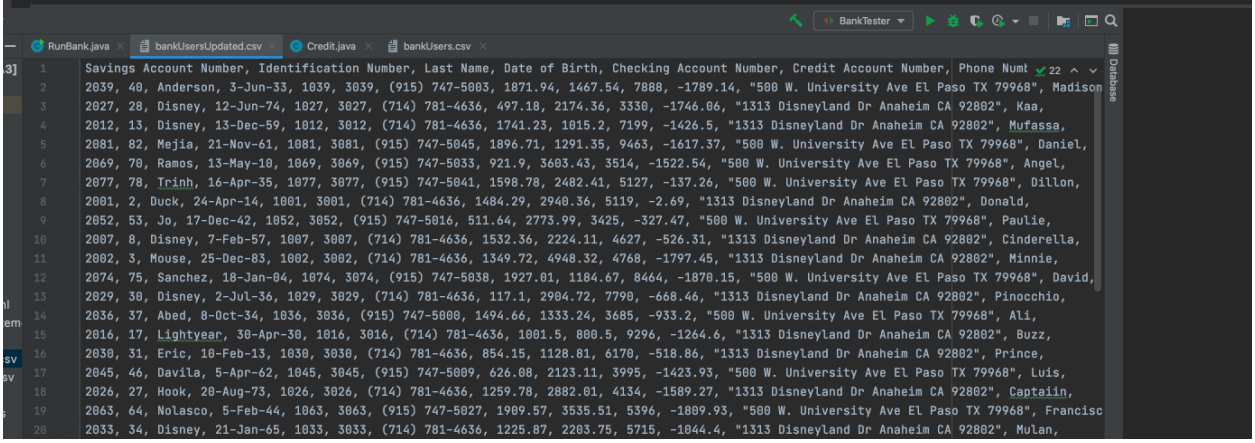
My test results in the Junit consisted of mostly of cases that would pass, in terms of having assertFalse for the incorrect way of doing things, and assertTrue for the correct way of doings things. A picture of that is included below. Also, in Junit testing I added messages so I could see if it was performing the transactions the way I wanted it to. For the other tests I did in my code, I have included screen shots of all my generated files and how they assisted in checking that everything was correct.




RunBank
RunBank.java
transactionLog.txt
bankUsersUpdated.csv
transactionActions.csv
Credit.java
bankUsers.csv
Mickey Mouse made a payment in the amount of $100.0 from account 1000, to Donald Duck. Mickey Mouse's new balance for account 1000
Mickey Mouse transferred $400.0 in cash from 2000 to 1000. Mickey Mouse Balance for 2000 is $60.98. Balance for 1000 is $ 1512.85
Mickey Mouse made a balance inquiry on 2000. Mickey Mouse balance is $60.98
Mickey Mouse withdrew $100.0 in cash from 1000. Mickey Mouse Balance for 1000 is : 1412.85
Mickey Mouse deposited $500.0 in cash from 2000. Mickey Mouse Balance for 2000 is : 560.98
Mickey Mouse transferred $300.0 in cash from 2000 to 3000. Mickey Mouse Balance for 2000 is $260.98. Balance for 3000 is $ -389.07000000000000
Ali Abed deposited $5000.0 in cash from 1036. Ali Abed Balance for 1036 is : 6494.66
Kevin Acosta deposited $5000.0 in cash from 1037. Kevin Acosta Balance for 1037 is : 5674.2
Antonio Aguirre deposited $5000.0 in cash from 1038. Antonio Aguirre Balance for 1038 is : 6102.03
Madison Anderson deposited $5000.0 in cash from 1039. Madison Anderson Balance for 1039 is : 6871.9400000000005
Kevin Aofia deposited $5000.0 in cash from 1040. Kevin Aofia Balance for 1040 is : 6384.98
Angel Armendariz deposited $5000.0 in cash from 1041. Angel Armendariz Balance for 1041 is : 6236.21
Nadia Atiyeh deposited $5000.0 in cash from 1042. Nadia Atiyeh Balance for 1042 is : 5454.39
Nicole Avila deposited $5000.0 in cash from 1043. Nicole Avila Balance for 1043 is : 6612.07
Andrea Carrera Pio deposited $5000.0 in cash from 1044. Andrea Carrera Pio Balance for 1044 is : 6452.68
Luis Davila deposited $5000.0 in cash from 1045. Luis Davila Balance for 1045 is : 5626.08
Vincent De La Torre deposited $5000.0 in cash from 1046. Vincent De La Torre Balance for 1046 is : 6125.8
Oswaldo Escobedo deposited $5000.0 in cash from 1047. Oswaldo Escobedo Balance for 1047 is : 6967.75
Ivan Espino deposited $5000.0 in cash from 1048. Ivan Espino Balance for 1048 is : 6125.11
Julio Garcia deposited $5000.0 in cash from 1049. Julio Garcia Balance for 1049 is : 5609.26
Mauricio Garcia Tovar deposited $5000.0 in cash from 1050. Mauricio Garcia Tovar Balance for 1050 is : 5746.12
Denise Gomez deposited $5000.0 in cash from 1051. Denise Gomez Balance for 1051 is : 5999.53
Paulie Jo Gonzalez deposited $5000.0 in cash from 1052. Paulie Jo Gonzalez Balance for 1052 is : 5511.64
Raul Guerrero deposited $5000.0 in cash from 1053. Raul Guerrero Balance for 1053 is : 6141.1
Matthew Herrera deposited $5000.0 in cash from 1054. Matthew Herrera Balance for 1054 is : 5351.01
Ethan Kish deposited $5000.0 in cash from 1055. Ethan Kish Balance for 1055 is : 6117.29
Diana Licon deposited $5000.0 in cash from 1056. Diana Licon Balance for 1056 is : 6140.14


RunBank.java
transactionLog.txt
AtiyehNadiaBankStatement.txt
bankUsersUpdated.csv
BANK OF OOP
1801 Hawthore St.
El Paso, TX, 79902
==========
Nadia Atiyeh Sun 03-14-2021
"500 W. University Ave El Paso TX 79968"
(915) 747-5006
10-Mar-32
"500 W. University Ave El Paso TX 79968"
Identification Number: 43
==========
CHECKINGS Account #: 1042
Starting Balance: $454.39
Deposit of $20.00 into account number 1042. Current Balance is $474.39
Withdrawal of $20.00 from account number 1042. Current Balance is $454.39
Ending Balance: $454.39
==========
SAVINGS Account #: 2042
Starting Balance: $1971.48
Deposit of $20.00 into account number 2042. Current Balance is $1,991.48
Ending Balance: $1991.48
==========
CREDIT Account #: 3042
Starting Balance: $-808.33
Deposit of $20.00 into account number 3042. Current Balance is $-788.33
Ending Balance: $-788.33

```
1. Inquire account by Name
2. Inquire account by type and number
3. Inquire all
4. Add Account
5. Write Bank Statement
6. Back to Main Menu
Answer: 4

To start we will need some information
Please input your Full Name:
Manolo Atiyeh
Please enter your Date of Birth:
04-02-2001
Please enter your Phone Number:
(915)253-2776
Please enter your Address
6041 Palo Alto Ave.
Enter your starting balance
200.00
Savings Account Created
Would you like to create another account (Yes or No)
```



```
Savings Account Number, Identification Number, Last Name, Date of Birth, Checking Account Number, Credit Account Number, Phone Numb
2039, 40, Anderson, 3-Jun-33, 1039, 3039, (915) 747-5003, 1871.94, 1467.54, 7888, -1789.14, "500 W. University Ave El Paso TX 79968", Madison
2027, 28, Disney, 12-Jun-74, 1027, 3027, (714) 781-4636, 497.18, 2174.36, 3330, -1746.06, "1313 Disneyland Dr Anaheim CA 92802", Kaa,
2012, 13, Disney, 13-Dec-59, 1012, 3012, (714) 781-4636, 1741.23, 1015.2, 7199, -1426.5, "1313 Disneyland Dr Anaheim CA 92802", Mufassa,
2081, 82, Mejia, 21-Nov-61, 1081, 3081, (915) 747-5045, 1896.71, 1291.35, 9463, -1617.37, "500 W. University Ave El Paso TX 79968", Daniel,
2069, 70, Ramos, 13-May-10, 1069, 3069, (915) 747-5033, 921.9, 3603.43, 3514, -1522.54, "500 W. University Ave El Paso TX 79968", Angel,
2077, 78, Trinh, 16-Apr-35, 1077, 3077, (915) 747-5041, 1598.78, 2482.41, 5127, -137.26, "500 W. University Ave El Paso TX 79968", Dillon,
2001, 2, Duck, 24-Apr-14, 1001, 3001, (714) 781-4636, 1484.29, 2940.36, 5119, -2.69, "1313 Disneyland Dr Anaheim CA 92802", Donald,
2052, 53, Jo, 17-Dec-42, 1052, 3052, (915) 747-5016, 511.64, 2773.99, 3425, -327.47, "500 W. University Ave El Paso TX 79968", Paulie,
2007, 8, Disney, 7-Feb-57, 1007, 3007, (714) 781-4636, 1532.36, 2224.11, 4627, -526.31, "1313 Disneyland Dr Anaheim CA 92802", Cinderella,
2002, 3, Mouse, 25-Dec-83, 1002, 3002, (714) 781-4636, 1349.72, 4948.32, 4768, -1797.45, "1313 Disneyland Dr Anaheim CA 92802", Minnie,
2074, 75, Sanchez, 18-Jan-04, 1074, 3074, (915) 747-5038, 1927.01, 1184.67, 8464, -1870.15, "500 W. University Ave El Paso TX 79968", David,
2029, 30, Disney, 2-Jul-36, 1029, 3029, (714) 781-4636, 117.1, 2904.72, 7790, -668.46, "1313 Disneyland Dr Anaheim CA 92802", Pinocchio,
2036, 37, Abed, 8-Oct-34, 1036, 3036, (915) 747-5000, 1494.66, 1333.24, 3685, -933.2, "500 W. University Ave El Paso TX 79968", Ali,
2016, 17, Lightyear, 30-Apr-30, 1016, 3016, (714) 781-4636, 1001.5, 800.5, 9296, -1264.6, "1313 Disneyland Dr Anaheim CA 92802", Buzz,
2030, 31, Eric, 10-Feb-13, 1030, 3030, (714) 781-4636, 854.15, 1128.81, 6170, -518.86, "1313 Disneyland Dr Anaheim CA 92802", Prince,
2045, 46, Davila, 5-Apr-62, 1045, 3045, (915) 747-5009, 626.08, 2123.11, 3995, -1423.93, "500 W. University Ave El Paso TX 79968", Luis,
2026, 27, Hook, 20-Aug-73, 1026, 3026, (714) 781-4636, 1259.78, 2882.01, 4134, -1589.27, "1313 Disneyland Dr Anaheim CA 92802", Captain,
2063, 64, Nolasco, 5-Feb-44, 1063, 3063, (915) 747-5027, 1909.57, 3535.51, 5396, -1809.93, "500 W. University Ave El Paso TX 79968", Francisc
2033, 34, Disney, 21-Jan-65, 1033, 3033, (714) 781-4636, 1225.87, 2203.75, 5715, -1044.4, "1313 Disneyland Dr Anaheim CA 92802", Mulan,
```