

King Saud University
College of Computer and Information Sciences
CSC111 – Fall 2020
Lab Final Exam

Question 1 (10 points):

Bank	7.5 points
- names: String[] - balances: double[] - nClients: int	0.25
+ Bank() + Bank(size: int) + findIndexByName(name: String): int + addClient(name: String, balance: double): boolean + deleteClient(name: String): boolean + deposit(name: String, amount: double): boolean + withdraw(name: String, amount: double): boolean + transfer(fName: String, tName: String, amount: double):boolean + transferToBank(f: String, toBank: Bank, t:String, a: double): boolean + display(name: String): void + findMax(): String	0.25 0.25 1 1 1 0.5 0.5 1 1 0.25 0.5

TestBank	2.5 points
+ main(): void * Print and loop menu * Option "Add a client" * Option "Close a client account" * Option "Deposit" * Option "Withdraw" * Option "Transfer" * Option "Transfer to another bank" * Option "Display balance" * Option "Find the richest client" * Option "Exit"	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25

The Class: Bank

As shown in the UML diagram, write the class Bank that has the attributes:

- **names:** An array containing the clients names (no duplicates).
- **balances:** An array containing the clients balances.
- **nClients:** the number of clients.

The methods of this class are:

- **Bank(size):** A constructor that sets the banks nClients to 0, initializes both arrays names and balances to be of size **size**.

- **findIndexByName(name)**: returns the index of **name** in the array **names**. If **name** is not found, return -1.
- **addClient(name, balance)**: adds **name** to the array of names and **balance** to the array of balances in the same index, then returns true. If there is no space in the arrays, it returns false. If there was a client with the same name, it doesn't add the client and returns false.
- **deleteClient(name)**: removes a client by replacing his/her name and balance with the last client's name and balance, then returns true. Returns false if **name** was not found.
- **deposit(name, amount)**: adds the **amount** to the balance of **name**, then returns true. Then prints the name, previous balance, and current balance. If **name** is not found, return false.

Ali was 1900.0, now 2000.0

- **withdraw(name, amount)**: takes the **amount** from the balance of **name**, then returns true. Then print the name, previous balance, and current balance. If **name** is not found or the balance is not enough, return false without changing the balance.

Ali was 1900.0, now 1000.0

- **transfer(fName, tName, amount)**: takes the **amount** from **fName**'s balance and adds it to **tName**'s balance, then returns true. Then prints the name, previous balance, and current balance for both clients. If either **fName** or **tName** are not found or the balance is not enough, return false without changing the balance:

Ali was 1900.0, now 1000.0

Ahmad was 100.0, now 1000.0

- **display(name)**: prints **name** information. If **name** is not found, print "Not found."

Ahmad 134567.88 SR

- **findMax()**: returns the name of the richest client in the bank. If the bank has no clients, returns null.
- **transfterToBank(fName, toBank, tName, amount)**: takes the **amount** from **fName**'s balance and adds it to **tName**'s balance in **toBank**, then returns true. If either **fName** or **tName** are not found or the balance is not enough, return false.

Ali was 1900.0, now 1000.0

Joe was 100.0, now 1000.0

The Main Class: TestBank

- **main**: the main method does the following:
 1. Create a **Bank** object **b1** with the size of 5.
 2. Create another **Bank** object **b2** with size of 2.
 3. Add the following clients to the bank b2:
 - 1- Mike, 1000
 - 2- Joe, 3000

4. Print the following menu:
 - 1- Add Client.
 - 2- Close a client account.
 - 3- Deposit.
 - 4- Withdraw.
 - 5- Transfer.
 - 6- Transfer to another bank.
 - 7- Display balance.
 - 8- Find the richest client.
 - 9- Exit.
5. When option 1 is selected "Add Client." Ask the user to enter the name and balance of the new client. Print "DONE" if completed or "ERROR" not completed.
6. When option 2 is selected "Close a client account." Ask the user to enter the name of the client you want to close his/her account. Print "DONE" if completed or "ERROR" not completed.
7. When option 3 is selected "Deposit." Ask the user to enter the name and amount of money to add to name's account. Print "DONE" if completed or "ERROR" not completed.
8. When option 4 is selected "Withdraw." Ask the user to enter the name and amount of money to take from name's account. Print "DONE" if completed or "ERROR" not completed.
9. When option 5 is selected "Transfer." Ask the user to enter the name to transfer from, the name to transfer to, and the amount. Then, print "DONE" if completed or "ERROR" not completed.
10. When option 6 is selected "Transfer to another bank." Ask the user to enter the name to transfer from (this client is from bank b1), the name to transfer to (this client is from bank b2), and the amount. Then, print "DONE" if completed or "ERROR" not completed.
11. When option 7 is selected "Display balance." Ask the user to enter the name of the client you want to check his/her account.
12. When option 8 is selected "Find the richest client." Display the name of the richest client or "No clients" if no clients are in the bank.
13. When option 9 is selected "Exit", print a goodbye message.
14. Make sure the menu loops until option 9 is selected.

Sample run:

```
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Ahmad 1000
DONE
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Ahmad 2000
ERROR
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Ali 100
DONE
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Khalid 3000
DONE
*****
```

```

* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Hesham 400000
DONE
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Saad 1500.5
DONE
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 1
Enter name and balance: Mansoor 900
ERROR
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *
* 4- Withdraw.                  *
* 5- Transfer.                  *
* 6- Transfer to another bank.*
* 7- Display balance.           *
* 8- Find the richest client.   *
* 9- Exit.                      *
*****
=> 8
Hesham 400000.0 SR
*****
* 1- Add Client.                *
* 2- Close a client account.    *
* 3- Deposit.                   *

```

```

* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 2
Enter name: Hesham
DONE
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 8
Khalid 3000.0 SR
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 2
Enter name: Hesham
ERROR
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 3
Enter name and amount: Ahmad 200000
Ahmad was 1000.0, now 201000.0
DONE
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *

```

```

* 6- Transfer to another bank.*
* 7- Display balance.          *
* 8- Find the richest client.  *
* 9- Exit.                     *
*****
=> 8
Ahmad 201000.0 SR
*****
* 1- Add Client.               *
* 2- Close a client account.   *
* 3- Deposit.                  *
* 4- Withdraw.                 *
* 5- Transfer.                 *
* 6- Transfer to another bank.*
* 7- Display balance.          *
* 8- Find the richest client.  *
* 9- Exit.                     *
*****
=> 4
Enter name and amount: Hesham 10
ERROR
*****
* 1- Add Client.               *
* 2- Close a client account.   *
* 3- Deposit.                  *
* 4- Withdraw.                 *
* 5- Transfer.                 *
* 6- Transfer to another bank.*
* 7- Display balance.          *
* 8- Find the richest client.  *
* 9- Exit.                     *
*****
=> 4
Enter name and amount: Ahmad 1000
Ahmad was 201000.0, now 200000.0
DONE
*****
* 1- Add Client.               *
* 2- Close a client account.   *
* 3- Deposit.                  *
* 4- Withdraw.                 *
* 5- Transfer.                 *
* 6- Transfer to another bank.*
* 7- Display balance.          *
* 8- Find the richest client.  *
* 9- Exit.                     *
*****
=> 4
Enter name and amount: Ahmad 123456789
Ahmad was 200000.0, now 200000.0
ERROR
*****
* 1- Add Client.               *
* 2- Close a client account.   *
* 3- Deposit.                  *
* 4- Withdraw.                 *
* 5- Transfer.                 *
* 6- Transfer to another bank.*

```

```

* 7- Display balance.          *
* 8- Find the richest client.  *
* 9- Exit.                    *
*****
=> 5
Enter from name, to name, and amount: Ahmad Ali 100
Ahmad was 200000.0, now 199900.0
Ali was 100.0, now 200.0
DONE
*****
* 1- Add Client.              *
* 2- Close a client account.  *
* 3- Deposit.                 *
* 4- Withdraw.                *
* 5- Transfer.                *
* 6- Transfer to another bank.*
* 7- Display balance.         *
* 8- Find the richest client. *
* 9- Exit.                    *
*****
=> 5
Enter from name, to name, and amount: Ahmad Hesham 200
ERROR
*****
* 1- Add Client.              *
* 2- Close a client account.  *
* 3- Deposit.                 *
* 4- Withdraw.                *
* 5- Transfer.                *
* 6- Transfer to another bank.*
* 7- Display balance.         *
* 8- Find the richest client. *
* 9- Exit.                    *
*****
=> 5
Enter from name, to name, and amount: Hesham Ali 200
ERROR
*****
* 1- Add Client.              *
* 2- Close a client account.  *
* 3- Deposit.                 *
* 4- Withdraw.                *
* 5- Transfer.                *
* 6- Transfer to another bank.*
* 7- Display balance.         *
* 8- Find the richest client. *
* 9- Exit.                    *
*****
=> 5
Enter from name, to name, and amount: Ahmad Ali 123456789
Ahmad was 199900.0, now 199900.0
Ali was 200.0, now 200.0
ERROR
*****
* 1- Add Client.              *
* 2- Close a client account.  *
* 3- Deposit.                 *
* 4- Withdraw.                *

```



```

* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 6
Enter from name (this bank), to name (other bank), and amount: Ahmad Joe 500
Ahmad was 199900.0, now 199400.0
Joe was 3000.0, now 3500.0
DONE
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 6
Enter from name (this bank), to name (other bank), and amount: Joe Ahmad 400
ERROR
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 7
Enter name: Ahmad
Ahmad 199400.0 SR
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *
* 5- Transfer. *
* 6- Transfer to another bank.*
* 7- Display balance. *
* 8- Find the richest client. *
* 9- Exit. *
*****
=> 7
Enter name: Hesham
Not found
*****
* 1- Add Client. *
* 2- Close a client account. *
* 3- Deposit. *
* 4- Withdraw. *

```

```
* 5- Transfer.                *
* 6- Transfer to another bank.*
* 7- Display balance.         *
* 8- Find the richest client.  *
* 9- Exit.                    *
*****
```

=> 9

Goodbye

Question 2 (5 points):

A) (3 Marks) Write a Java program that will prompt the user to enter **n** student's names. Each student name is composed of first name and last name.

1. Enter the number of students.
2. Read and store those names in two different arrays: first and last.
3. Print all students.
4. Count and print how many students with first name "Mohammed". (Ignore case) (1 Mark)
5. Remove all students that have a last name that starts with 'AL' (Ignore case) (You can remove with: remove and shift, or copy last).

Hint: You can use CharAt(i) to access certain characters of a string, or use substring(int1,int2) to retrieve a portion of a string. (2 Marks)

6. Print all students.

Call your class **Names**.

Sample run:

```
Please enter how many students: 6
Fahad Mohammed
Ali Abdullah
Mohammed AlFahad
Mohammed Abdullah
Saad alqahtani
Hamad AlHamad
The students names are:
Student 0 : Fahad Mohammed
Student 1 : Ali Abdullah
Student 2 : Mohammed AlFahad
Student 3 : Mohammed Abdullah
Student 4 : Saad alqahtani
Student 5 : Hamad AlHamad
There are 2 Students with first name "Mohammed"
The students with last name not beginning with "AL" are:
Student 0 : Fahad Mohammed
Student 1 : Ali Abdullah
Student 2 : Mohammed Abdullah
```

B) (2 Marks) Write a Java program that will read and store a series of integers input by the user ending with zero in an array **nums** (Number of integers must not exceed 100).

1. Print the array **nums**.
2. Prompt the user to enter two integers **j** and **k** where $j < k$, both are not negative, and both are less than the number of entered integers (You must check those two conditions and if any of them not true you must display an appropriate message and ask the user to try again).
Print all the array elements between index **j** and index **k** (including **j** and **k**). (1 Mark)
3. Create another array **nums2** of the same size as **nums**, and copy all the elements that are negative in **nums** to **nums2** (you don't have to delete them from **nums**). (1 Mark)
4. Print **nums2**.

Call your class **Numbers**.

Sample run:

```
Enter The Numbers: 3 88 4 2 -3 7 -12 -66 8 -14 56 0
The numbers are: 3 88 4 2 -3 7 -12 -66 8 -14 56
Please enter two locations j and k: 5 2
ERROR: j must be greater than k
Please enter two locations j and k: -2 5
ERROR: j and k can't be negative.
Please enter two locations j and k: 5 150
ERROR: j and k must be less than the number of entered numbers.
Please enter two locations j and k: 2 5
The elements between 2 and 5 are: 4 2 -3 7
The elements of nums2: -3 -12 -66 -14
```