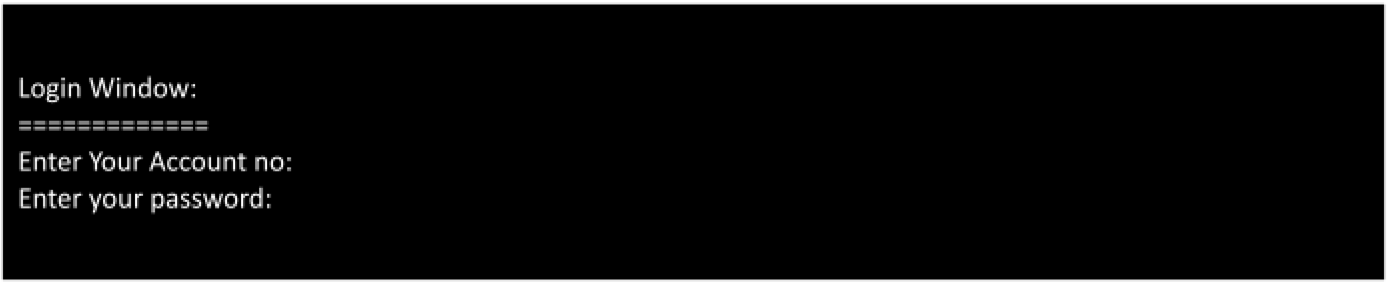
ATM Machine

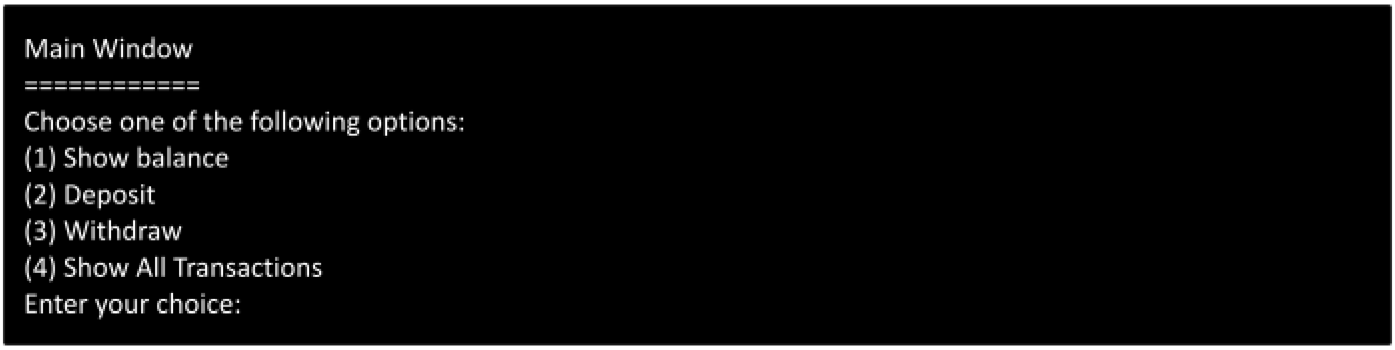
In this assignment you will create a program that simulates an ATM machine using C++. The proposed program should display the following screen to the user (Figure 1):



```
Login Window:
=============
Enter Your Account no:
Enter your password:
```

(Figure 1 – Login Window)

The user should enter his/her credentials in order to see the second screen:



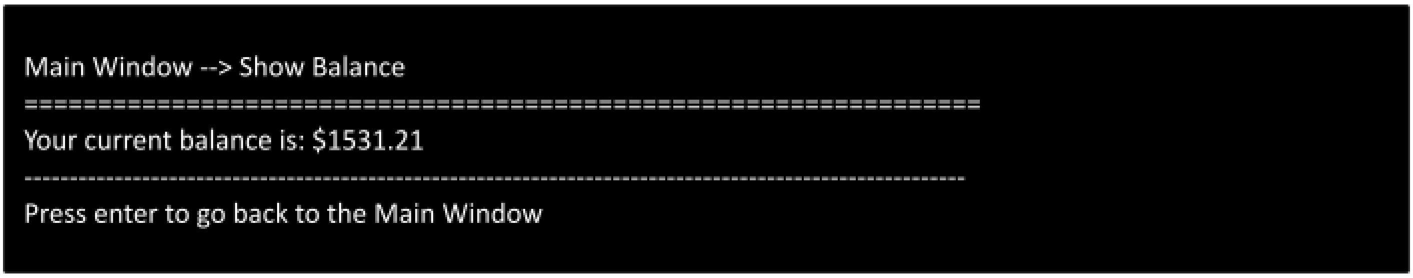
```
Main Window
============
Choose one of the following options:
(1) Show balance
(2) Deposit
(3) Withdraw
(4) Show All Transactions
Enter your choice:
```

(Figure 2 - Main window)

As soon as we have the user choice, one of the following windows should appear to the user:

Show Balance:

This should display the client current balance, and should enable the user to get back to the main window.

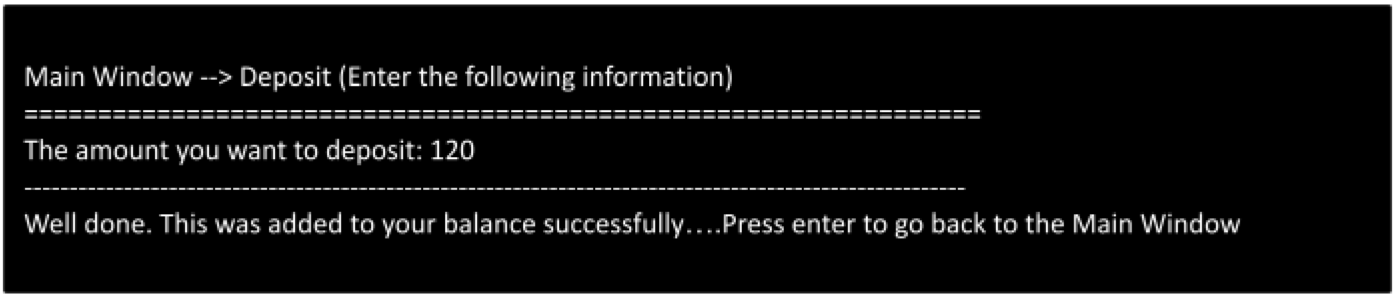


```
Main Window --> Show Balance
=================================================================
Your current balance is: $1531.21
----------------------------------------------------------------------------------------------------
Press enter to go back to the Main Window
```

(Figure 3 – Show balance window)

Deposit:

This option is initiated when the user chooses (2) in the main window. The user has to enter the amount he/she wants to deposit. The new deposit should be added to the account's balance and the user should be able to get back to the main window. The program should verify that this is a valid positive number.

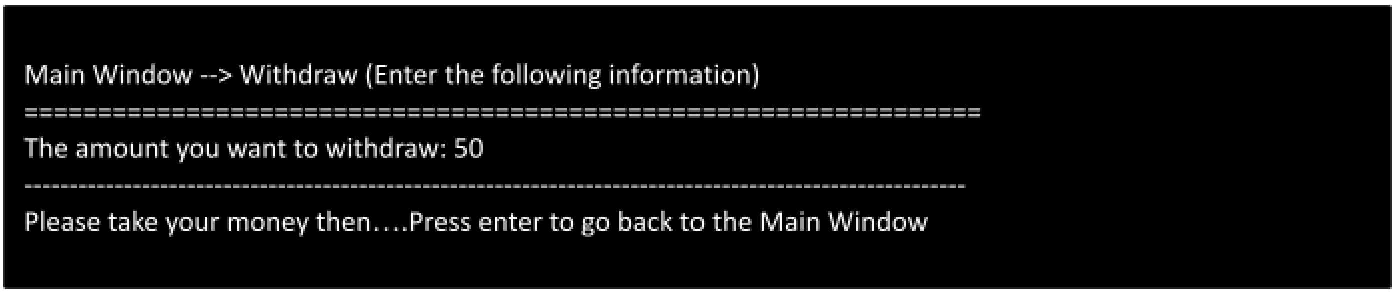


```
Main Window --> Deposit (Enter the following information)
==================================================================
The amount you want to deposit: 120
----------------------------------------------------------------------------------------------------
Well done. This was added to your balance successfully....Press enter to go back to the Main Window
```

(Figure 4 - Deposit window)

Withdraw:

This option is initiated when the user chooses (3) in the main window. The user has to enter the amount he/she wants to withdraw. The transaction should be completed only if the amount requested by the user is a valid positive number that does not exceed the client's current balance, the account's balance should be updated accordingly and the user should be able to get back to the main window.

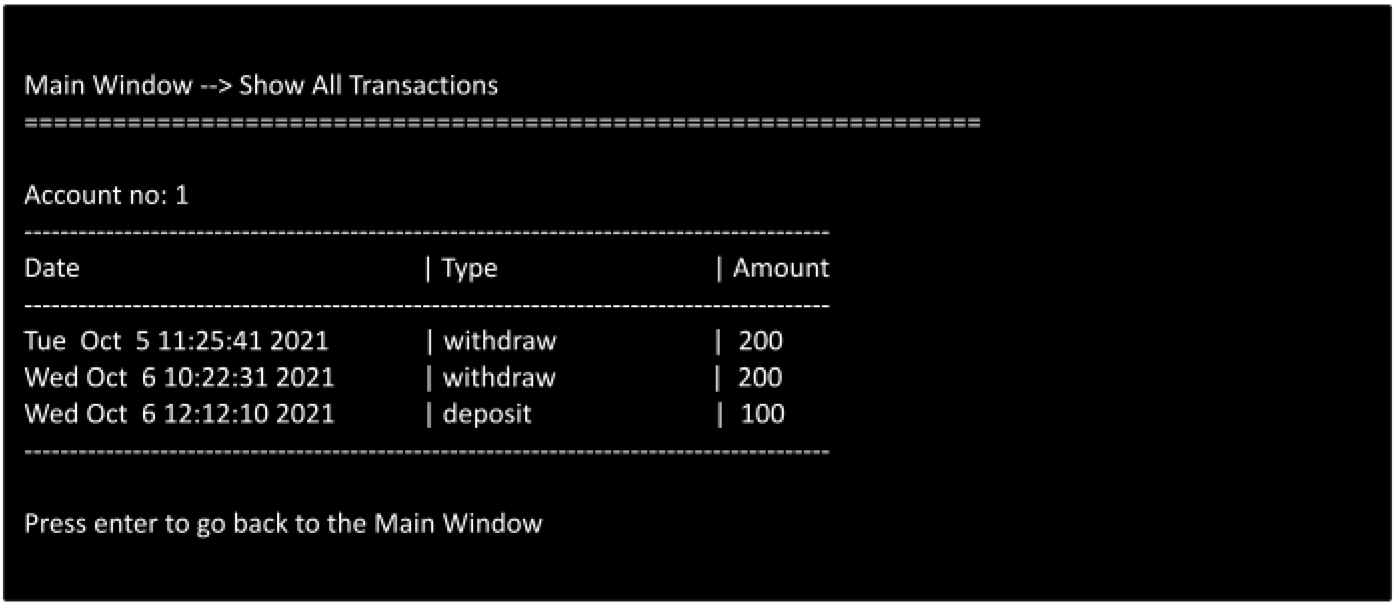


```
Main Window --> Withdraw (Enter the following information)
==================================================================
The amount you want to withdraw: 50
----------------------------------------------------------------------------------------------------
Please take your money then....Press enter to go back to the Main Window
```

(Figure 5 - Withdraw window)

Show All Transactions:

This option is initiated when the user chooses (4) in the main window. The program will display the transactions made by this account (figure 6). and the user should be able to get back to the main window.



```
Main Window --> Show All Transactions
==================================================================

Account no: 1
-------------------------------------------------------------------------------------
Date                             | Type                   | Amount
-------------------------------------------------------------------------------------
Tue  Oct  5 11:25:41 2021        | withdraw               |  200
Wed Oct  6 10:22:31 2021         | withdraw               |  200
Wed Oct  6 12:12:10 2021         | deposit                |  100
-------------------------------------------------------------------------------------

Press enter to go back to the Main Window
```

(Figure 6 - Show All Transactions window)

Important Notes:

- There is no need to save the data of the accounts in any file (The main function should start by creating an array (or vector) of 10 bankAccounts (initialize these accounts with the password (123) and 5000 as a balance) and data will not be saved when the user exits the program.

- Be sure to represent each account as an object that has (account_no, password, balance, array of transactions).
- Each transaction has: date(text), type (“withdraw” or “deposit”) and an amount.
- The account_no. should be assigned automatically everytime we create any new account, starting from 1 (i.e. the first created account has account_no 1, the second created account has account_no 2, the third created account has account_no 3, etc.). Do NOT use the size of the array (vector) to generate the account_no (this should be generated by the Account class constructor).
- It is so important to submit a working program (Non-working applications will not be considered).
- Be sure to include the code of all required classes and methods.
- You must submit one file only (a zip file that includes all your code).