## **Program 1 – ATM Machine**

In this assignment you will create a program that allows a user to do the following:

1. Create a bank account by supplying a user id and password.
2. Login using their id and password.
3. Quit the program.

Now if login was successful the user will be able to do the following:

1. Withdraw money.
2. Deposit money.
3. Request balance.
4. Quit the program.

If login was not successful (for example the id or password did not match) then the user will be taken back to the introduction menu.

**Assignment 2 – Part 1**

**Able to implement** **C++ Basic & flow control**

* C++ Basic: Done
* Flow control: Done
* Complete Program: Based on requirements

**Able to implement C++ concepts exception handling, Header files, comments**

* Exception handling: Multiple Examples Done
* Header files: 5 Header Files Done
* Comments: Constantly Throughout

**Able to create reusable components like printIntroMenu() & printMainMenu**

* Used reuseable components: Yes
* What are they called:
  + void printIntroMenu();
  + void login();printMainMenu();
  + void depositMoney();
  + void widthdrawl();
  + void bankBalances();
  + void createAccount();
  + void wipeData()

**Able to give Classroom Presentation & prepare GitHub README File**

* Updated Readme File: Thursday, October 24, 2024
* Presentation: After passing in assignment

GitHub: <https://github.com/Fall2024-NSCC-ECampus/assignment-2-console-applications-atm-machine-cadalac-don.git>

**Able to implement to all given requirements, and create feasible business logic**

This is a basic ATM program. This program can be modified for more complicated ATM devices. The logic flow for this ATM code can be expanded for transferring money to various account like Checking and Savings. There are other many suggestions to be added to.

There are different ways we could have created the interface approached. I used a single-input approach but I could have the user input all the info at once with one prompt before doing comparisons. There are many “right” ways of doing this.

Technically not required based on requirement outcomes but these could be great next steps:

* We could use string arrays for multiple variable storage and have a function to start the info
  + This would be useful if we were to handle multiple account
  + However, not ideal to have multiple accounts stored locally to an ATM machine
  + Otherwise, a remote secure database could serve this better
* Wiping data after
  + A time interval of last action
  + After quitting creating an account or logging in
* You cannot login if you do not create account first