

Please follow the naming convention for both the email subject and file name.  
(CMPS301\_Project5\_GroupName)

## Project 5: Banking System program

You will write a program that will help in the management of a private bank.

### General Requirements

- Write a program that reads (each time the program is executed) from the secondary storage from a file named accounts.txt the accounts information into an array (named accounts) of type struct (named accountType). The array can store up 1000 elements.
- Your program will ask the user to choose one of the following options:
  1. Deposit Money
  2. Withdraw Money
  3. Transfer Money
  4. Show Transactions
  5. Open an Account
  6. Close an Account
  7. Sort Accounts from largest balance to smallest
  8. Sort Accounts in alphabetical order using the account owners name
  9. Quit Program
- The program will have at least ten user-defined functions (one function for each of the eight options 1-8, one to print the options menu, and one to find the index of the account in the array).
- The program will save the transactions information in a separate file that has the same name as the account number (e.g. for account number: 15689 the file name will be 15689.txt)
- If the user tries to withdraw more than the available balance an error message will be displayed and the transaction will not be processed.
- Closing an account will only flag the account rather than permanently delete the account.
- If the user tries to close a non-zero balance account an error message will be displayed and the transaction will not be processed.
- When the user decides to quit the program the array is saved back to the secondary storage to the same file (i.e. accounts.txt)
- The first function's name is printOptionsMenu, which will print (show on the screen)

-----  
Please choose one of the following options:

1. Deposit Money
  2. Withdraw Money
  3. Transfer Money
  4. Show Transactions
  5. Open an Account
  6. Close an Account
  7. Sort Accounts from largest balance to smallest
  8. Sort Accounts in alphabetical order using the account owners name
  9. Quit Program
-

- The second function's name is `getArrayIndexOfAccount`, which receives three parameters (the account number, the array of accounts, and the length of the array) and returns the index of the account in the array if found; otherwise; the function returns -1.
- The third function's name is `depositMoney`, which receives four parameters (the index of the account in the array, the amount to be deposited, the array of accounts, and the length of the array). The function will update the balance in the array and will save (by appending) the transaction information (including the date and time) in the file that has the same name as the account number.
- The fourth function's name is `withdrawMoney`, which receives four parameters (the index of the account in the array, the amount to be withdrawn, the array of accounts, and the length of the array). The function will update the balance and will save (by appending) the transaction information (including the date and time) in the file that has the same name as the account number.
- The fifth function's name is `openAccount`, which receives two parameters (the array of accounts, and the length of the array passed by reference). The function asks for the account owner's information (e.g. name and opening balance). The function will add a new record to the array of accounts and create a new file (with the same name as the account number) and update the transactions information (including the date and time) with the opening balance.
- The sixth function's name is `closeAccount`, which receives two parameters (the array of accounts, and the length of the array) which checks the balance before flagging the account as deleted. If the balance is not zero, an error message will be displayed and the transaction will not be processed.
- The seventh function's name is `sortBalances`, which receives two parameter (the array of accounts and the length of the array). The function will sort and display the accounts (without changing the original array) from largest balance to smallest.
- The eight function's name is `sortNames`, which receives two parameter (the array of accounts and the length of the array). The function will sort and display the accounts (without changing the original array) in alphabetical order using the account owner's name.
- Make sure to follow the instructions carefully including using the exact function names with the exact letter cases (upper and lower case letters). Your programs are going to be tested using a unified main function that is developed for that purpose. If the unified main function does not compile this indicates inconsistencies in meeting the above requirements **and will result in a zero grade.**
- All functions (except the main function) will be defined as follows:
  1. The function prototype is placed before the main
  2. Comment describing what the function does appears directly under function prototype
  3. Formal variable names are not used in any prototype
  4. The functions definition is placed after the main

Additional Requirements:

- Be sure to comment your code adequately.
- Be sure to indent properly. Check the textbook to see how it should be done.
- Use self-documenting variable names

Sample contents of the accounts.txt file:

Serial	Account	FirstName	LastName	Balance	Flag
1	56897	Richard	Dustin	1001.59	0
2	43534	Michael	Smith	123.55	0
3	65743	George	Gonzales	10.66	0
4	43346	Donald	Malik	0	1
5	97867	Owen	Evelyn	5527.10	0
6	38531	Lucas	Wyatt	7891.22	0