UNIVERSITY of NICOSIA

COMP-113 Programming Principles II

Spring 2022

Instructor: Dr Athena Stassopoulou

Project 2

OUT: Thursday, April 7 10, 2022.
DUE: Thursday, May 5, 2022 (source file on moodle, hard-copy in class).

No assignments will be accepted after this deadline.

INSTRUCTIONS:

1) Submit:
   a) The .cpp file by uploading it at the course site (.cpp file ONLY). PLEASE name the file with your name as follows: AndreasAndreou.cpp

   b) A hard copy of your code with the corresponding output screen printed. (NO LOOSE SHEETS OF PAPER WILL BE ACCEPTED. Submit everything neatly in a folder).

2) Your program should work with functions which should be provided to perform the various tasks.

   You may only use the C++ programming concepts covered so far in class.

   The examiner will attach great importance to the quality of code. Marks will be deducted for poor programming style. The program should be commented and indented to aid clarity.

   NOTE: This is an individual assignment and should represent your own ideas, work and effort. In cases of plagiarism, a grade zero will be assigned to all parties involved.

We want to create part of a system for a small bank. The information to be stored for each customer should include the following (use structs):

- *Name*
- *Address*
- *ID number (e.g. L8112990)*
- *Accounts: Assume one customer may have more than one account up to a pre-defined maximum of 5. Each account has a unique number, a type of account (assume for simplicity three types only: checking, savings and money market account) and a balance.*
- *Number of Accounts maintained by customer*

The database of customers is to be implemented by an array of size *max*, where *max* is a suitable constant (e.g. 100).

Provide a program including functions for the following tasks (the more you modularize your program the better it will be):

1. **Read a customer**: Read the information of a new customer, making sure the ID entered for the new customer is not already in use. The customer should be inserted at the end of the array, just after the last customer. This function should also enable an opening of accounts for the new customer. You should therefore ask the user how many accounts he/she would like to open and enter the appropriate account details (by using function 6 below).
2. **Return the number of customers:** Return how many customers are in the array.
3. **Find a customer:** Write a function which takes the customer's ID number as parameter and returns true if there is a customer with that ID number and false otherwise.
4. **Print Account holder details:** Write a function which takes the account number as parameter, and lists the details (name, address and ID number) of the account holder.
5. **Print Customer details:** Write a function that takes the customer ID as parameter and prints: name, address and details of all accounts held by this customer.
6. **Open a new account:** Open a new account for a customer, given the name and customer ID number as parameters. Your function should make sure that the number of accounts of a customer does not exceed the pre-defined maximum (i.e. 5). When opening a new account, all account details should be provided. Note: The account number given must be unique and should not belong to any other customer. The type of account should be entered and it should be one of three types: checking, savings or money market account. The money market account requires a minimum balance, higher than the other two types, in order to open and maintain it. (Set this minimum to a constant number of your choice).

7. **Delete customer:** Delete a customer given the customer ID number as parameter. On deletion the number of customers in the array should be updated and the array should be changed so that there are no gaps between consecutive cells. (This operation is allowed only if the total balance of a customer is zero. The appropriate message should be given to the user).
8. **Withdraw amount:** Given the customer name, customer ID and an amount as parameters, make a withdrawal of that amount of money from an account of the customer's choice (The function should present the account numbers to the user and allow him/her to make a choice). If the account chosen does not have enough balance then the user should be notified of the balance and be allowed to re-enter an amount for withdrawal. (Remember that for money market account type, the minimum balance should be always maintained).
9. Provide reports for the following requirements:
   (a) Percentage of customers with savings accounts.
   (b) Percentage of customers with more than three accounts.
   (c) Details of the customer with the maximum total balance. (Total is the sum of all account balances of that customer).