**cs3307 – Object oriented analysis and design**

**Assignment 1: Introduction to C++**

[**Banking System Project**](http://www.cppforschool.com/project/banking-system-project.html)

**Introduction:**

You are to write a program (a banking system) in C++. In order to keep focus on learning the language, the system will have text-based menus. By all means, if you want to program a graphics user interface then I will not be a hurdle in your learning. ☺

The following is a description of the approximate system so as to give you an idea of the system. You are to make it precise by documenting specific requirements the system will implement.

**System description:**

The bank has an unspecified number of customers. For demonstration purposes, you can have a manageable number of customers. The term “client” is a synonym for customer. Each customer has either or both of a savings (S) account and a chequing (C) account.

There is a bank *manager* who has managerial powers to open and close an account and see the critical details of a particular, or all, customers in a formatted display.

A *customer* has power on to use *only* his/her account within the permissible operations. Note that a customer can use multiple operations in one session of system use.

Each user of the system, regardless of his/her role type, has an “id” for secure login. After logging into the system, s/he will be bounded by his/her role as to what s/he can do with this system.

The banking system has the following core features (remember this is approximate!):

* Open/Close an account; close restricted to zero balance accounts.
* Deposit and withdraw money from an account.
* Give warning messages for not sufficient funds (e.g., when withdrawing).
* Transfer a sum from one account to another.
* Obtain account balances.
* Give a warning message to the client if the balance on his/her “chequing” account will drop down below the threshold of $1,000 – prior to him/her executing the operation. If s/he decides to go ahead despite the warning message, a charge of $2.00 is levied on the client for each such transaction.
* A client can open either savings or chequing account, or both; s/he can add the complementary account at anytime.
* The bank manager can display the account details of any given customer, or all customers, and obtain aggregate data on the funds in the bank, etc., categorised appropriately.

There is also a role of a systems *maintenance person* who also has an id to access the system. S/he can turn ON/OFF execution trace. If the trace is ON, then the system will dump the execution traces in a file that can be printed out at a later time by the maintenance person. Each execution trace will log user identification data, time of system access, and the execution trace (which system functions are entered and exited) for the operation conducted by the user. Thus, the system will dump execution traces, one per customer use of the system, in a sequence bounded by ON/OFF trigger.

Enhance the above given requirements as you deem fit; bonus marks will be considered. ☺

**Project requirements:**

1. Deliverable 1: Document textually the requirements, categorised by roles, that have been implemented and are demonstratable. Each requirement should be uniquely identifiable with an “id”.
2. Deliverable 2: Identify requirements that are beyond the core that you have created.
3. Deliverable 3: Write the program in C++. Deliver source code along with any instructions on how to compile and run the program and any system configuration parameters.
4. Deliverable 4: Document five scenarios in a descriptive form. For each scenario, show using screen dumps the behaviour of the system as observable from the outside.
5. Deliverable 5: Please describe what you have learnt by doing this assignment, including what you would do differently if you were to do this assignment again.
6. All documents should be readable on a Windows machine with standard application programs.
7. Submission format: In a zipped folder. The folder name *must* follow the following format:
   * <”Assigment” assignment #><underscore ”Group” group # underscore><hyphenated last names of group members>
     + E.g., Assignment 1\_Group 5\_Bloggs-Carpenter
8. Deadline: 24th September midnight
9. Recipient email address to be decided.

Have fun!