CSE 1325-001 Homework #5 – Model, View, Controller

In this homework assignment, you will be re-structuring your previous assignment (HW4) to fit the Model, View, Controller (MVC) Software Pattern. Also, we will be incorporating the work we did in HW2 by incorporating files.

Part 1: MVC

For this part, you will need to re-structure your HW4 to fit MVC. This will be based off the full-credit implementation. You may use the sample solution on Blackboard as a base if you do not wish to use your own HW4 solution. You will need to modify some the existing functions in the Model to not use cout, but return values dependent on the function’s purpose. All output statements must be handled by the Controller and the View classes.

You will also need to modify the UML from HW4 to fit HW5. The HW4 UML is included in the HW4 solution. This homework will introduce the concept of design, so you can build the Controller and View classes however you see fit. Also, if you wish to change anything in the existing classes, you may do so. Just make sure any changes you make are present in the modified UML.

Part 2: Files

For any system, being able to save and load data from a file is important. That is what you will be doing for the Transaction Management System.

You will need to add Save and Load functionalities to your program. The save file will contain the contents of the list of transactions. When the file is loaded, it will read in the contents, create any necessary objects, and add them to the list. When the file is saved, it will overwrite the contents of the save file, if the file already exists. If the file does not exist, it will create the file and save to it. The name of the save file must be “abc1234_save_file.txt”. You may format the save file however you want.

Any changes for this part must be shown in your modified UML.

Bonus (5pts)

For the Files portion of this homework, you must override the >> operation so that the contents of the save file directly into the contents of the Transaction_List, Trasaction, and Date objects. By this, I mean that “ist >> date;” is a valid operation, where ist is a input stream and date is a Date object. Make sure this is updated in your UML diagram.

Deliverables

You will submit your code and screenshots via Blackboard. You will upload a zip file, named "abc1234_HW5zip", which contains 1 folder (2 if you did the bonus)

- full_credit
  - abc1234_HW5_Class_Diagram.xmi
  - abc1234_Date.h and abc1234_Date.cpp
  - abc1234_Transaction.h and abc1234_Transaction.cpp
  - abc1234_Transaction_List.h and abc1234_Transaction_List.cpp
  - abc1234_View.h and abc1234_View.cpp
  - abc1234_Controller.h and abc1234_Controller.cpp
  - abc1234_main.cpp
  - abc1234_save_file.txt
  - makefile
  - abc1234_main.png (or multiple if multiple screenshots were taken). These screenshots will be picture of your code running in terminal.
  - Instructions for compiling and running your code (either in comments in blackboard or in a README file)
- bonus_1
  - abc1234_HW5_Class_Diagram.xmi
  - abc1234_Date.h and abc1234_Date.cpp
  - abc1234_Transaction.h and abc1234_Transaction.cpp
  - abc1234_Transaction_List.h and abc1234_Transaction_List.cpp
  - abc1234_View.h and abc1234_View.cpp
  - abc1234_Controller.h and abc1234_Controller.cpp
  - abc1234_main.cpp
  - abc1234_save_file.txt
  - makefile
  - abc1234_main.png (or multiple if multiple screenshots were taken). These screenshots will be picture of your code running in terminal.
  - Instructions for compiling and running your code (either in comments in blackboard or in a README file)

Full credit files named incorrectly result in a loss of 5 points each.