**COP 4331- Fall 2014**

**Homework 2**

**Due: October 16, 2014**

* **No late submissions will be accepted**
* **Please submit the homework through Webcourses using the following format: COP4331C\_Fall14\_HW#\_LastName\_FirstName.pdf, where # will be 1, 2, or 3.**
* **This homework assignment is an INDIVIDUAL assignment, not a team effort!!!**

**Problem statement:**

You are required to improve the system developed in Homework-1. You have been requested to improve admin functionality to view student’s activity. Currently, admins cannot view which student registered to which courses and the time each student spent online for each course. The admins should be able to search students with longest and shortest times spent on each course, number of videos the students watched and the newest students in the system. In order to improve the system you have been asked to force request the following requirements when storing student information: <FirstName>, <LastName >, <CourseName>, <RegisterDate>, <TimeSpent> and <NumberOfVideosWatched>. Using this information an automated sorter will generate a list of the top 3 students according to sorting criteria.

The categories are as follows:

1. Register Date (Sorted by students who lastly or firstly registered to a course)
2. Time Spent (Sorted by students who spent longest or shortest time for a course)
3. Number of videos watched (Sorted by students who watched the most or least number of videos)

The search can be done by selecting a course and a sorting criteria (search option).

**Assignment:**

1. You will be provided a .txt file that **you must use** for your submission. This .txt file will essentially be a “database” of courses and students. It will be named “input.txt”. Your program must automatically open and run “input.txt”.
2. Your program should rank students based on the above categories (6 search options in total).
3. Keep in mind that while your program should work for the sample input; it should **also work** for different input files that follows the same format.

**Deliverables:**

* All submissions must be a zip file that contains the source code, .exe files and README file including information necessary to run your programs (Hint: think of the readme file as a preliminary attempt at creating a user manual).
* Your program should compile and run. If it doesn't compile and/or crash, you will receive only a partial credit for the code. **IT MUST RUN USING THE GIVEN “INPUT.TXT” FILE.**
* You have two options for language, C++ and Java. If you choose C++, you have two options, you can either submit a makefile or you can submit it as a Visual Studios Project. If you choose Java, you **must** submit the Jar, as well as the source code, or submit an Eclipse/NetBeans Java project. If there are issues with these things, you can let us know and we’ll try to work with you, but only good reasons for exceptions will be considered.

**Constraints of input.txt:**

* The first line in input.txt will be a number from 1 – 100 representing the number of courses

Following that, will be ‘n’ number of courses in the following format:

* Course Name: The name of the course that the student is registered
* FirstName LastName: The first name and last name of a student who registered to the course (there is a space between the first name and last name)
* Register Date in format: MM/DD/YYYY
* Time spent in minutes: The total time spent by the student for this course (an integer number)
* Number of videos watched: (an integer number)

Example:

1

Introduction to Computer Science

Alice Taylor

9/15/2014

324

3

Bob Fisher

8/13/2014

511

8

**Output:**

Handling output is entirely up to you. You will be providing a “User Manual” in the form of a README.txt file so you are not constrained to an output method.

Examples for ways to handle output could be:

1. User Driven, prompting the user to select the sorting criteria they want to use and show the top 3 students according to the selected criteria
2. Automatic, running your program either dumps to the screen or to a separate text file the results of analyzing the input file.