**COP 4331- Spring 2013**

**Homework 2**

**Due: February 21, 2013**

* **Please submit the homework through webcourses2 (please don’t send files via email!)**
* **Homework is individual assignment, not to be worked on as a group!**
* **No late submissions will be accepted**

**Problem statement:**

You are required to improve on the system developed in Homework-1. You have been requested to improve on the existing friend feature, and add a suggested friend feature. Currently, users have to manually search for students located within their classes and using the limited personal information provided, determine if they would like to send them a friend request. In order to improve the system you have been asked to force the following requirements when creating a personal profile: Current classes, GPA, Major, Current year (ie: Freshman, Sophomore, Junior, Senior). Using this information an automated sorter will generate a list of the top 3 recommended friends based on: Current Classes, GPA, Major, and Current Year. Suggested friends should obviously share a class with the user, those that do are ranked using first their GPA, followed by if they share majors with the user, and lastly ranked by their Current year.

Assume there is an algorithm already in place that reads through potential friend profiles and reduces them down to the relevant information, and then return it in the form of an array to be checked if they are potential highly compatible friends.

**Assignment:**

1. Correct/expand the class diagram of HW1 to now contain Student, Major, GPA, Years, and an algorithm that accepts and sorts applicants. Be sure that everything that appears in the description can be seen at the diagram.
2. Implementation of all the classes in the class diagram (using an Object Oriented language)
3. Documentation following a coding standard (e.g. Java Coding Standard)
4. Implementation of an example scenario

 \*Example user looking for friends:

 - Name: John Doe

 - Current Classes: COP 4331, CDA 3103, PHY 2049, MAC 2312

 - GPA: 3.5

 - Major: Computer Science

 - Current Year: Junior

 \* 4 potential friends (with the important features based on their profile, extracted by another algorithm you do not have to implement)

 - Name: John Dorian

 - Current Classes: COP 4331, CDA 3103, PHY 2048, MAC 2313

 - GPA: 3.75

 - Major: Computer Engineering

 - Current Year: Junior

 - Name: Christopher Turk

 - Current Classes: COP 4331, CDA 3103, PHY 2049

 - GPA: 3.75

 - Major: Computer Science

 - Current Year: Senior

 - Name: Elliot Reid

 - Current Classes: EEL 3657, EEL 3041, EEL4440

 - GPA: 4

 - Major: Electrical Engineering

 - Current Year: Sophomore

 - Name: Glenn Matthews

 - Current Classes: COP 4331, COT 4210, COP 4020, COP3402

 - GPA: 3.8

 - Major: Computer Science

 - Current Year: Junior

Output should be the properly ranked ordering of the 3 potential friends, in this case Glenn Matthews, followed by Christopher Turk, followed by John Dorian.

**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.
* Your program should compile and run. If it doesn't compile and/or crash, you will receive only a partial credit for the code.
* 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 can submit the Jar, as well as the source code, or submit a 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.

**Hints:**

* The java coding standard can be found at: <http://java.sun.com/docs/codeconv/>.
* Implement the User\_Database as an array of Users with the above important elements, and create a function that accepts the array of users and an individual user as it's input. Your function should pass through the user array and remove all users who don't share a class with that given individual. Then it should rank them on GPA, by Major, and by Year (with the ordering that Seniors get the highest rank).