Project 2

In your next project, you have to implement sorting algorithm to sort databases of any type. There are two databases you need to consider - students and employees. You have already implemented students database in project1. You have to create a similar database for employees. There are two categories of employees - faculty and staff. The common attributes are id, name, salary, and entrance year. Besides, faculties have an attribute that represents number of supervised students and staffs have an attribute that denotes their department. You have to create classes to store the employee database. Next, you need to implement one sorting algorithm - bubble sort. The bubble sort algorithm is a well-known sorting algorithm. You can find an explanation of the sorting algorithm along with pseudocode in the link here (http://en.wikipedia.org/wiki/Bubble_sort).

Sorting criteria is different for students and employees. You need to sort students according to GPA (in a non-ascending order). If GPA of two students are same, you need to sort according to names of the students (alphabetically). If both the criteria are same for two students, the order does not matter. For employees, you need to sort records according to their salary (in a non-descending order). If salary of two employees are same, you need to sort according to their entrance year (in a non-descending order). Again, if both the criteria are same for two employees, the order does not matter.

So, for this project, you have two databases of student and employee and one sorting algorithm bubble sort. Your have to design your code in a way so that user can apply any sorting algorithm on any of these data types.

Inputs:
Two separate input files will be provided - one for students and one for employees. In the first input file, each line starts with either ‘U’ or ‘G’ to represent undergraduate or graduate student followed by a list of attributes to initialize one student. The input ends when it encounters ‘Q’ at the start of a new line. Similarly, in the second input file, each line starts with either ‘F’ or ‘S’ to represent faculty or staff followed by a list of attributes to initialize one employee. Two sample files are provided herewith.

How to run the program:
the program will run as follows
program_name first_input_file second_input_file

Output:
Your program should output to the screen (i.e. stdout). The output of the program will be sorted lists of students and employees. A sample output file is provided herewith.

Requirements:
1. You MUST use abstract factory pattern to design your program.

1. Your design should be able to accommodate any new sorting algorithm without any change in the existing classes.
2. Your design should be able to accommodate any new data types without any change in the existing classes.
3. All attributes of a class must be private.
4. Any repetitions should be avoided.
5. Your program must compile, run, and produce output similar to sample output.
6. You can use any drawing tool to draw your UML diagram. UML diagram should be in jpg, png, or pdf format.
7. Please avoid multiple submission.

Handin:
1. A UML class diagram of your project clearly mentioning the attributes and methods of each class.
2. All of your program files (*.h, *.cpp) along with Makefile
3. A text file named team.txt clearly explaining the individual responsibilities of each member of the team.

Deadline:
This project is due via “handin”(http://secure.cse.msu.edu/handin/) by **11:59pm on Feb 12, 2014**. Make sure to include all required files as mentioned.