OOP RETAKE EXAM

The Computer Science Department is evaluating 5 professors to see which professor has the highest rating according to student input. You will generate a **ProfessorRating** class consisting of professor name and the ratings for 4 aspects: *Easiness*, *Helpfulness*, *Friendliness* and *Clarity*. The value for each rating is in the range of 1 to 5, with 1 is the lowest and 5 is the highest.

Your program should contain the following functionality:

Write a class named ProfessorRating with 5 data members: profName with string type, and 5 **dynamically allocated arrays** for the marks obtained for each of the aspects: Easiness, Helpfulness, Friendliness and Clarity.

Your class should contain the following methods:

1. constructors (default and copy constructor) and assignment operator.

2. destructor.

3. overload of the insertion operator : operator>> to display a professor's name and **average** ranking for each of : *Easiness*, *Helpfulness*, *Friendliness* and *Clarity.*

4. void addRanking(int easy, int help, int friendly, int clear) – appends the rankings for each one of the 4 aspects: *Easiness*, *Helpfulness*, *Friendliness* and *Clarity.*

5. double calcAverageRating() –

((average(easiness)+average(helpfulness)+average(friendliness)+average(clarity)/4.0)

In your main program, generate a vector of ProfessorRating which holds a list of 5 professor ratings.

const int NUM_PROFESSORS=5;

vector<ProfessorRating> csProfs(NUM_PROFESSORS);

Write a function to locate the professors with the highest and lowest average rating and display the name and rating of the professors. Call the function from your main() program to output the information.

void displayHighestandLowestRating(vector<ProfessorRating> profList)

Write a function to display the average of ALL ratings of the professors using the class's calcAverageRating () function. Call this display function from your main() program to output the information.

void displayAverageRating(vector<ProfessorRating> profList).

You don’t need to follow the **model -view -controller** architecture. It is ok if you have just one class and the main.cpp source file.