# Classes and Data Abstraction with Pointers

* Indent code and insert comments according to coding standard guideline to document your program.
* Program must be implemented and run as instructed.

Student class and ClassRoom class implementation from the main function file and, in addition, the Student class and ClassRoom class must have its interface (Student.h/ClassRoom.h) in a separate file from its implementation (Student.cpp/ClassRoom.cpp).

The code will be written in the following files:

* Student.cpp and ClassRoom.cpp
* Student.h and ClassRoom.h
* StudentMain.cpp
* Please use these file names exactly.

Include executable and input data file

Write a program that creates Student Objects using data from the attached “students.txt” file. Add code to limit the number of students created to 24. At a minimum, each Student object should contain the following attributes: **(26 pts)**

* First name (string and const) - (5 pts)
* Last name (string and const) - (5 pts)
* Social Security Number (string) – (4 pts)
* Four Exam Grades (array of type double) – (4 pts)
* Student Average Exam Grade (double) – (4 pts)
* Student Number (int and static) – Initially set to zero. Each time a Student object is created, the Student Number is incremented by one and when destroyed, decremented by one. (4 pts)

Add member functions to the Student class to: **(9 pts)**

* Function to return the average Exam Grade of a student: (Sum of Four Exam grades divided by four).
* Getter and Setter Functions for each attribute
* Function to display associated attributes of the Student

Create a single ClassRoom object with the following attributes **(9 pts)**

* Name of the ClassRoom object (string ) – Choose any name (i.e. CSC134)
* Count of the number of students in the ClassRoom (int)
* Dynamic Array of Student Objects (maximum size is 24) – Represented as an array of pointers.

Add member functions to the ClassRoom class to: **(32 pts)**

* ClassRoom Constructor Function – (4 pts)
* ClassRoom Destructor Function – (4 pts)
* Read the input data file and create the Student objects. – (4 pts)
* Sort list of students by student average – (4 pts)
* Sort list of students by student last name – (4 pts)
* Return the average grade of all students (Sum of Student Average divided by number of students). – (4 pts)
* Display a summary of all the students in the ClassRoom object and the associated attributes of each student attributes. – (4 pts)
* Return the count of student objects created – (4 pts)

Write the main() driver to demonstrate the following: **(24 pts)**

1. Create a ClassRoom Object (3 pts)
2. Use the ClassRoom Object to read the Student Objects from the input file ”students.txt” (3 pts)
3. Use the ClassRoom Object to sort list of students by student average (3 pts)
4. Display the list of students by student average (3 pts)
5. Use the ClassRoom Object to sort list of students by student last name (3 pts)
6. Display the list of students by student last name (3 pts)
7. Use the ClassRoom Object to display the average grade of all students. (3 pts)
8. Use the ClassRoom Object to display the number of Student objects created (3 pts)

**NOTE: No global non-constant variables should be used. You can add more data members or member functions for Student and ClassRoom Class Definitions if needed.**

Alfalfa Aloysius 123-45-6789 90.0 100.0 83.0 49.0

Alfred Francis 123-12-1234 97.0 96.0 97.0 48.0

Gerty Gramma 567-89-0123 80.0 60.0 40.0 44.0

Android Alexis 087-65-4321 23.0 36.0 45.0 47.0

Bumpkin Fred 456-78-9012 78.0 88.0 77.0 45.0

Rubble Betty 234-56-7890 90.0 80.0 90.0 46.0

Noshow Cecil 345-67-8901 81.0 65.0 49.0 43.0

Buff Bif 632-79-9939 20.0 30.0 40.0 50.0

Airpump Andrew 223-45-6789 75.0 90.0 100.0 83.0

Backus Jim 143-12-1234 85.0 97.0 96.0 97.0

Carnivore Art 565-89-0123 71.0 80.0 60.0 40.0

Dandy Jim 087-75-4321 92.0 23.0 36.0 45.0

Elephant Ima 456-71-9012 19.0 78.0 88.0 77.0

Franklin Benny 234-56-2890 50.0 90.0 80.0 90.0

George Boy 345-67-3901 40.0 11.0 91.0 84.0

Heffalump Harvey 632-79-9439 30.0 91.0 20.0 30.0