The assignment will be graded out of 100 points.

**Task 1 (100 pts.)**

File [compute\_grades.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/compute_grades.java) contains an incomplete program. The goal of the program is compute the semester grade for each student in a class, based on exam and homework scores for that student. The grades are stored in a CSV file, following this format:

* The top row contains column names.
* After that, each row contains data for a specific student.
  + Column 0 contains first names.
  + Column 1 contains last names.
  + Columns 2, 3, 4 contain exam scores.
  + Columns 5 to 11 contain homework scores.

An example of such a CSV file is [data1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/data1.txt).

Complete that program, by defining a process\_grades function, that satisfies the following specs:

* The function takes three arguments, called input\_name, csv\_name, pretty\_name.
* Argument input\_name is the name of the CSV file containing the grade data that your code needs to process.
* Argument csv\_name is the name of an output CSV file, where your code will save the grade data that you have computed. This output CSV file will have a top row with column names, and then one row per student. The file will have five columns:
  + Column 0 contains the name of the student (concatenation of first name and last name, separated by space).
  + Column 1 contains the average exam score for that student.
  + Column 2 contains the average homework score for that student.
  + Column 3 contains the smallest value between the exam average score and the homework average score for that student.
  + Column 4 contains the letter grade, by converting the value at column 3, using standard threshold (90 and above is A, 80 to 90 is B, 70 to 80 is C, 60 to 70 is D, below 60 is F).

An example output CSV file following these specifications is [csv1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/csv1.txt).

* Argument pretty\_name is the name of an output text file, where your code will print out the grade data that you have computed, in a nice, tabular format that is easy for humans to read. This output text file will have a top row with column names, and then one row per student. The file will have the same five columns as the output CSV file described above. However, each column will be aligned. To obtain aligned columns, my solution uses this code to print each row:
* out.printf("%20s: %10.2f, %8.2f, %9.2f, %s\r\n",
* name, exam\_score, hw\_score, min\_score, grade);

An example output CSV file following these specifications is [pretty1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/pretty1.txt).

For example, suppose that:

* You use this [data1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/data1.txt) as your input file.
* You enter "csv1.txt" for the CSV output file, and "pretty1.txt" for the pretty-print file.

Then, your code should produce files that look EXACTLY like files [csv1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/csv1.txt) and [pretty1.txt](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment11/pretty1.txt) obtained using my solution. Use the same format, column names and spacing as you see in these example output files.

**IMPORTANT: you are only allowed to modify the provided code by writing the requested function, and possibly additional auxiliary functions (you are encouraged to come up with several such auxiliary functions in this task). You are NOT allowed to modify in any way the main function, or any other function that is already provided.**

**Suggestions**

Pay close attention to all specifications on this page, including file names and submission format. Even in cases where the program works correctly, points will be taken off for non-compliance with the instructions given on this page (such as wrong file names, wrong compression format for the submitted code, and so on). The reason is that non-compliance with the instructions makes the grading process significantly (and unnecessarily) more time consuming. Contact the instructor or TA if you have any questions.

**How to submit**

The assignment should be submitted via [Blackboard](http://elearn.uta.edu). Submit a single java file, called compute\_grades.java.