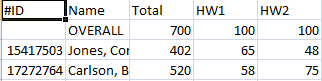
# CptS 233 PA #1

At certain universities, some professors teach large classes in which an individual TA is responsible for grading a given section using Excel. Assuming that each TA fills in the Excel form correctly (a rather large assumption), the instructor must combine each section's grades into a master gradebook file that can be published to students. Less technical (and in some cases lazy) professors combine the gradebook by hand. However, this process can be cumbersome and is most certainly error prone. Your task for this assignment is to automate this process by writing a Java application that automatically combines several CSV-based gradebooks into a single master gradebook.

## CSV File Format

Each CSV contains a single section's grades for a given category (e.g. homework). All files have a similar format. The top row contains headers and the 2nd row contains the maximum grade for each assignment. Similarly, the first column is always the student's ID number, the 2nd the student's name, and the 3rd is their overall score for the grading category. Rows after the third row contain marks for a given task. Here's an example:



### Obtaining a list of CSV files

Your program should accept a list of CSV files from the command line (using argc and argv). Each file will be in the format "<category>\_<number>.csv". For example: "homework\_1.csv", "homework\_2.csv", and "exams\_2.csv" are examples of potential file names. **Note that <category> can be any string**. Listed previously are strings that make sense, but your homework should be able to handle something like "sdfsrwe\_1.csv" just fine! Also, be sure to use <category> as the separator for the summary CSV file (discussed later).

## Expected Output

Your program should generate two CSV files: one summary file with overall grades and one details file that contains the grading breakdown for all sections.

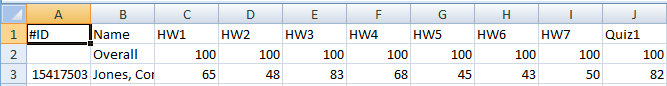
### Summary CSV File

The summary CSV file should contain all students, an overall class grade (in percent), and a points breakdown for each grading category. Here's an example screenshot of a summary file whose individual CSV files had a "Homework" and "Quizzes" <category> as inputs:



### Details File

The details file should essentially aggregate all of a student's records into a single file. In other words, it should include a detailed breakdown of every student's marks in the individual CSV files. Here's an example:



## Deliverables

You must commit your program to github repository no later than due date specified on Blackboard.

## Grading Criteria

Your assignment will be judged by the following criteria:

* [75] Test cases (25pts / ea). For each test case, your program should generate a complete and accurate summary and details CSV file.
* [5] Data structure usage. Your program utilizes the data structures discussed thus far in class. At minimum, your program must use hash tables and ~~vectors~~/arraylists.
* [10] Your code is well documented and generally easy to read.
* [10] Your program intelligently uses classes when appropriate and generally conforms to good OOP design (i.e. everything isn't slapped into main).