

## CS/CE 1337 – PROJECT 3 – Exam Grader

**Pseudocode Due:** 2/18 at 11:59 PM

**Project Due:** 3/4 at 11:59 PM

**Submission:** All programs are to be submitted in eLearning. The pseudocode should be submitted as a Word or PDF document. Please submit the program as a .cpp file only. Projects submitted after the due date are subject to the late penalties described in the syllabus. Programs must compile and run in Visual Studio 2010. Pseudocode will not be accepted late.

**Problem:** You have been hired by a school district to write a program that will grade multiple choice exams and provide a report listing the grade (numeric and letter) for each student, the problems missed for each student (including the student's answer and correct answer) as well as a summary for the entire class giving the mean, median and mode of the exams. All arrays used must be created dynamically and only pointer notation may be used to interact with the arrays.

**Input:** The correct answers for the exam will be in a file named *answers.txt*. The first line of the file will be the number of questions to grade (maximum 100). Each subsequent line will contain the answer to sequential questions on the exam. You do not have to validate the exam answers file. It will contain exactly as many answers as indicated on the first line of the file, and all answers in the file will be a letter A-E.

The student exams will be in a file named *exams.txt*. Each line of the file will contain data for a unique student. Each line of the file will begin with a student ID (a 10-digit number – can be treated as a string) followed by the student's answers for the exam. Each answer will be separated by a space and there will be exactly as many answers as there are questions listed in *answers.txt*.

**Output:** All numeric values should be rounded to 2 decimal places. Letter grades follow the traditional grading system (no plus-minus, A=90 – 100, B=80-89.99, etc.). Numerical grades are calculated by dividing the number of correct answers by the total number of answers and converting to a percentage.

A file named *exam\_report.txt* should be created to hold the output from grading the exams. For each student, the following information should be written to the file (each bullet represents a separate line):

- Student ID
- <Numeric grade> <space> <letter grade>
- Exam questions missed (numeric - separated by spaces)
- Incorrect answers given by student (line up each answer with the ones digit of the question number in the previous line)
- Correct answer (line up each answer with the ones digit of the question number in the previous line)
- Blank line

After each student report has been generated to the file, create a summary for the exam stating the mean, median and mode of the exams. If there is more than one mode, then list all modes separated by a comma and a space (e.g. 65, 70, 75). Please follow the format displayed in the sample output file for the statistical summary.

**Functions:** Your program should contain functions to do each main task. There should be minimal code in the main function other than function calls. Do not chain your functions together (one function calls the next function which calls the next function, etc.)

A sample of each type of file is attached to the project listing in eLearning.