

## CS 215 – Spring 2021 – Lab 3

### Correction published Feb 17 8:00am

#### Learning Objectives:

- more practice with formatted user input and output
- writing formatted data to a text file
- use of a Sentinel Controlled Loop to solve a problem
- use of the Summation Loop Strategy to solve a problem

#### General Description:

Write a program that asks the user for data on courses for a semester, calculates the GPA and writes a report to a file as shown below.

First, the program opens a file for writing called “grades.txt”. When the file fails to open, a message “Unable to open file grades.txt” is printed and the program ends immediately.

The program next asks the user to enter the student’s name, which may have spaces. It then writes the “header” portion of the report.

The program then uses a Sentinel Controlled Looping strategy, asking the user to enter a Course ID (no spaces) or “done” to quit entering more data. It is assumed the user enters data for at least one course.

When “done” is entered, the looping ends.

When a Course ID is entered, the program:

- asks the user to enter the (credit) *hours* for the course
- asks the user to enter the letter *grade* for the course
  - o You may use only the first character of the user’s input, and upper case it:  
`char grade = toupper(inputGradeString[0]);`
- calculates the *Quality Point Value* of the letter grade:
  - o A=4, B=3, C=2, D=1, anything else = 0
- calculates the *Quality Points Earned* for the course (QPV \* Hours)
- keeps a sum of *Total Quality Pointes Earned* and *Total Hours Earned*

After “done” is entered, the GPA is calculated (TotQPE / TotHours) and the “footer” is written to the output file as shown in the example.

Finally, the program closes the output file, prints “Grade report written to grades.txt” and ends the program.

**Formatting:** the formatting of the user interaction on the Console should look as it does in the sample below, including blank lines and lining up of inputs. The formatting of the output file should look as it does in the example output file.

### Sample Execution:

```
Microsoft Visual Studio Debug Console
Enter student name:      Ada Byron
Enter Course (or done): CS215
Enter number of hours:  4
Enter grade (A,B,C,D,E): A

Enter Course (or done): KHS101
Enter number of hours:  1
Enter grade (A,B,C,D,E): E

Enter Course (or done): CHEM101
Enter number of hours:  4
Enter grade (A,B,C,D,E): B

Enter Course (or done): ART222
Enter number of hours:  2
Enter grade (A,B,C,D,E): D

Enter Course (or done): EGR101
Enter number of hours:  1
Enter grade (A,B,C,D,E): ?

Enter Course (or done): EGR102
Enter number of hours:  2
Enter grade (A,B,C,D,E): C

Enter Course (or done): done
Grade report written to grades.txt
```

### Sample (matching) output file:

Updated Feb 17 8:00am

```
GRADE REPORT FOR: Ada Byron
Course   Grd Hrs QPts
-----
CS215    A   4   16
KHS101   E   1    0
CHEM101  B   4   12
ART222   D   2    2
EGR101   ?   1    0
EGR102   C   2    4
-----
Overall:      14   34  GPA: 2.43
```

### Grading Rubrics:

- standard comment box at the top of the code.
- opens output file and checks for failure correctly.
- asks for student name and writes report header to file correctly
- asks for course: ends processing when **done** entered; continues processing when other is entered.
- asks for hours and grade correctly
- calculates QPV, QPE, Total QPE, Total Hours and GPA correctly
- writes report footer correctly
- closes file, writes ending message, and returns 0 at the end of the program

**Submit:** your .cpp file after demonstrating to the TA.