IT 226 Project 3: Data Integration

Start: 12th November 2014 Due:3rd December 2014 11:55pm

1. This is a group project. Only one submission per project is required.
2. All members of the group will get the same grade on the project.
3. This project must be completed using only C++. You must ensure that your project compiles and runs correctly on our Linux machines.

In this project, you will write a program that combines data from one or more grade books from various courses at our School. This combined data will be stored suitably, and can be "sliced" in several ways to export required data. You will provide a simple text-based interface to add or retrieve data.

Data

Three grade books from three courses have been provided to you. These contain simulated data, but in a format acceptable to Reggienet. Each file is basically an exported version of an Excel spreadsheet. The data format is a .csv file (comma separated values). They are all text files, so open them in Notepad to see their contents.

1. Each line represents one row of a spreadsheet which is all the data for one student in that course. Within a row, all columns are separated by commas. Note that column data may contain several words separated by spaces, but commas are reserved only as separators.
2. The first line of each file contains the column headings. Some special column headings that will be important for you will be:
    a. "User ID" or "Student Id": these will be ULIDs and will be unique for each student.
    b. If a column heading contains the string "comment" (case insensitive), then the contents of that column are pure text. Note that this may show up as "Comment : Assignment 1", or "Assignment 1 comments".
    c. If a column heading contains the string "grade" (case insensitive), then the contents of that column are letter grades. Note that this may show up as "grade", "Grade", "Letter grade", etc.
    d. If a column heading contains the string "name" (case insensitive), then the contents of that column are the first, last or complete names of students.
    e. All other columns contain individual grades for each assignment (as a number out of 100) or the cumulative total (also as a number out of 100).
3. The second line onwards contains the data, one student per line. The data for a student is arranged in the same order as the column headings.

The provided data has "three dimensions": student (identified by Id), semester and year (identified as "Fall" and "2005"), and course (identified as course number "IT 226"). When you are provided a file by the user, you are also provided explicitly which semester, year and course this file corresponds to.

What to do

1. Write classes that act as a repository of data read from all the provided files, such that:
    a. It is possible to extract data for a single student across courses and semesters.
    b. It is possible to extract data for a single semester, across students and courses.
    c. It is possible to extract data for a single course, across students and semesters.
2. To organize the data as efficiently as possible (in space as well as to complete the above operations), you have the following data structures at your disposal (you are not expected to create your own data structure):
    a. STL vector

    b. STL list
    c. STL set
    d. STL map
    e. Regular arrays
3. Write a class that reads the data in the csv format, extracts data from it and using the column headings, organizes it. Lastly it must add it to the existing repository of data.
4. Provide a simple text-based interface that presents a keyboard-key driven menu to the user and allows the following operations:
    a. Add data ('a' or 'A'):
        i. Take file name from the user.
        ii. Take semester and year from the user (e.g. "Fall" and "2005")
        iii. Take course number from the user (e.g. "IT 226")
        iv. Read the provided file (if it exists), extract the data and add it to the repository along with the provided course number, semester and year information.
        v. Print the number of students whose data it just read, and how many students already existed in the repository.
    b. Save data for a student ('s' or 'S'):
        i. Take student ID (e.g. "ashesh"). Note that the student ID may have numbers.
        ii. Take the name of the file where data must be exported.
        iii. Find all data from the repository pertaining to this student, and export it in csv format as follows:
            1. Column headings: "Student Id", <Course>-<Semester>-<Year>-<Assignment name>,…repeated for each instance found for this student.

            For example, if data was found for this student in two courses: IT 226 in Fall 2002 (which has 5 assignments and a letter grade) and IT 279 in Spring 2003 (which has 3 assignments and a letter grade), then the columns would be:
            Student Id,IT226-F-2002-Assignment 1, IT226-F-2002-Assignment 2, IT226-F-2002-Assignment 3, IT226-F-2002-Assignment 4, IT226-F-2002-Assignment 5, IT226-F-2002-Letter grade IT226-S-2003-Assignment 1, IT226-S-2003-Assignment 2, IT226-S-2003-Assignment 3, IT226-S-2003-Letter grade
    c. Exit the program ('e' or 'E').

Please make sure to check that your program does not crash with invalid input (i.e. a file name that does not exist, etc.). You can assume that the provided file will always be a csv file in the format explained above.

Help

1. Look at the stringstream class and the getline(istream,string,delimiter) function to see how you can read a line and separate it using the ',' as your delimiter.
2. Look at the documentation of the STL classes at cplusplus.com.
3. Some student IDs that repeat across some of the provided files: 3xr9yx, 1wvs78, u439zz5. Open these files in excel to verify the data for these students.

Grading

The following will be roughly the grading rubric:

1. Parsing the files correctly (40 points).

2. Organizing the data (25 points).
3. Text interface and saving to files (25 points).
4. Presentation (10 points)

Presentation

During the last week of class, each group will present to the class the details of their project. This is worth 10 points.

1. The presentation should be no more than 10 minutes.
2. The presentation should be made using slides (i.e. not just talking).
3. Not all group members are required to talk, but all group members are required to stand with the presenter.
4. Your presentation should include the following aspects of your project:
    a. How do you organize your data? Why did you choose this way?
    b. Which data structures did you use to organize your data and why.
    c. A short demo of your program.
5. Grading:
    a. Content (point 4 above): 4 points
    b. Visual aids (how neat and informative are your slides and demo): 3 points
    c. Presentation style (do you target the audience of your peers, do you talk like you don't care?): 3 points

A note about Git on Linux

Git on linux isn't working yet. For the time being, transfer files to windows before you commit/checkout from the repository. This is a poor workaround, and hopefully we can get Git to work on linux this week.

What to submit

Each group of each team should submit the following:

1. All the source code files (*.h, *.cpp) and a Makefile.
2. Each group will submit a group evaluation form.
3. Each member must submit a peer evaluation form.

All submissions must be made on Reggienet. No email submissions allowed!