CS250 Intro to CSII
Array of Structs Lab

Implement a database of Pacific Students using C++. The database will be read from a file called "enrolled.txt" and there are at most 1000 students in the database. The data file is organized as follows for each student:

FirstName LastName PUNetID GPA

Each piece of data is separated by a space and no spaces are found in the FirstName, LastName, or PUNetID. The GPA is a number of the form: #.##

1) Declare a struct Student that is able to hold the information for a student.

2) Create an array in main capable of holding 1000 Students.

3) Use the keyboard to read in one student. This user will enter the student's information using the same format as the file. Insert the student into the first element of the array (this student does not appear in the file). Use the student listed here as the student to be entered from the keyboard:

John Doe Doe1234 3.8

4) Write a function that prints out a single student as below. Test your function.

Doe1234: John Doe GPA: 3.8

5) Write a function that reads in a single student from the file into a struct. Test your function.

6) Write a function that displays a single student's information. Test your function.

7) Write a function that reads in all of the students from the file into the array of students. This function will call the function in step 5. Test your function. This function will start writing data into slot 0 of the array, overwriting the data from step 3 above.

8) Write a function that will display all of the students in the array to the screen. Test your function. This function must display two students per line, separated by a tab.

BONUS
8) Write a function that accepts the array of students and a GPA. Print all of the students in the database with a GPA greater than or equal to the GPA passed in. Test your function.

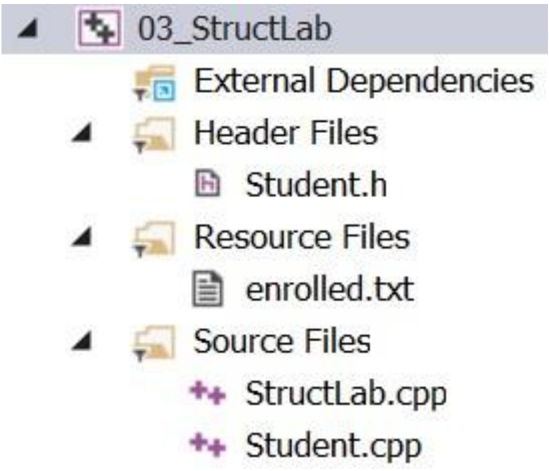
NOTES:

Build a project 03_StructLab in your InClass solution.

You will build a header file (Student.h) to contain the struct and the function prototypes for the Student operations.

You will build a source file (Student.cpp) to contain the function definitions for the Student operations.

You will build a source file (StructLab.cpp) to contain the main() function and project specific functions.



```
◢ 03_StructLab
     External Dependencies
  ◢  Header Files
       Student.h
  ◢  Resource Files
       enrolled.txt
  ◢  Source Files
       StructLab.cpp
       Student.cpp
```

DOUBLE BONUS:

9) A simple non-efficient sorting routine is given below. I would like you to write a sorting function that will sort your array of Student by GPA from lowest price to highest. Test your function.

```
void bubbleSort (float values[], int size)
{
  bool bAnySwitches = true;
float  temp;

  while (bAnySwitches)
  {
    bAnySwitches = false;

    for (int i = 0; i < size - 1; ++i)
    {
      if (values[i] > values[i + 1])
      {
        temp = values[i];
values[i] = values[i + 1];
values[i + 1] = temp;
bAnySwitches = true;
      }
    }
  }
}
```