# CSC 225

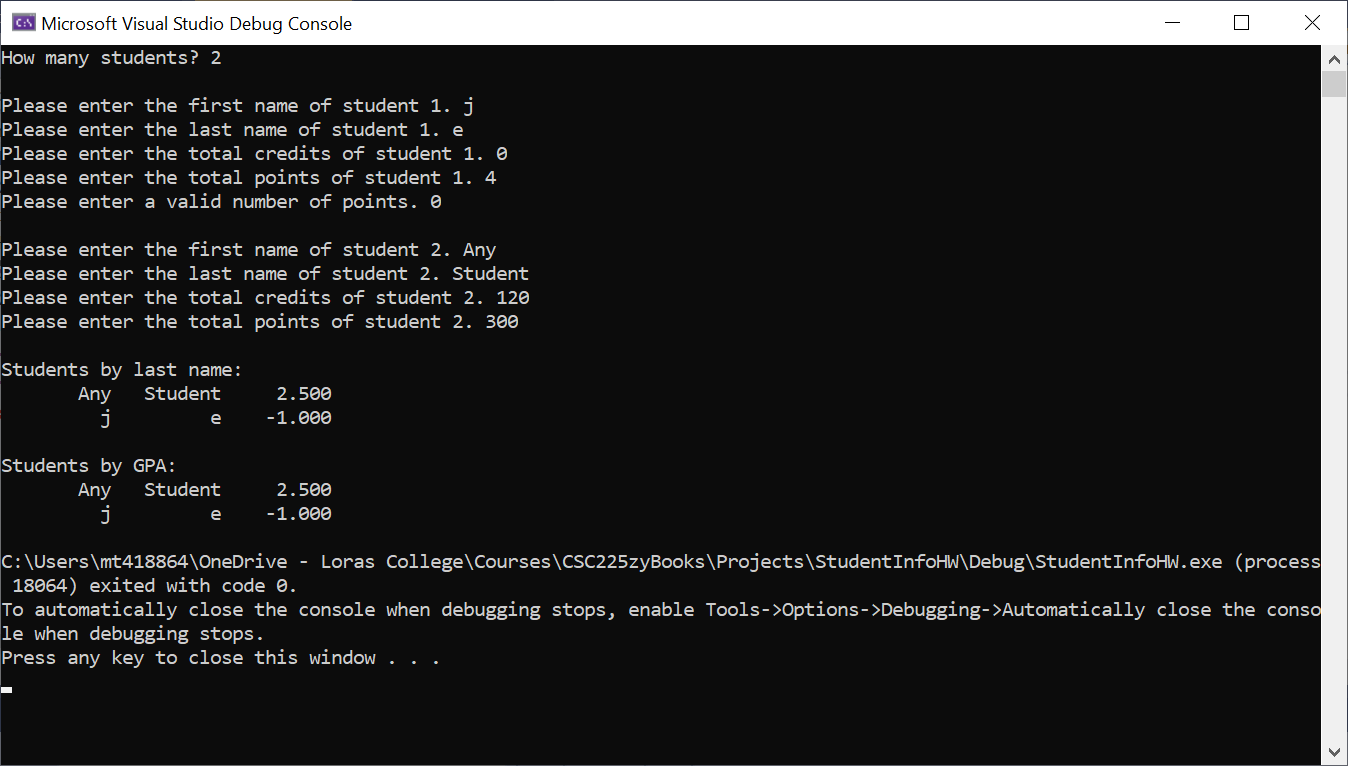
**Homework 1**

Due: Wednesday, October 5, 2022 at 11:59pm

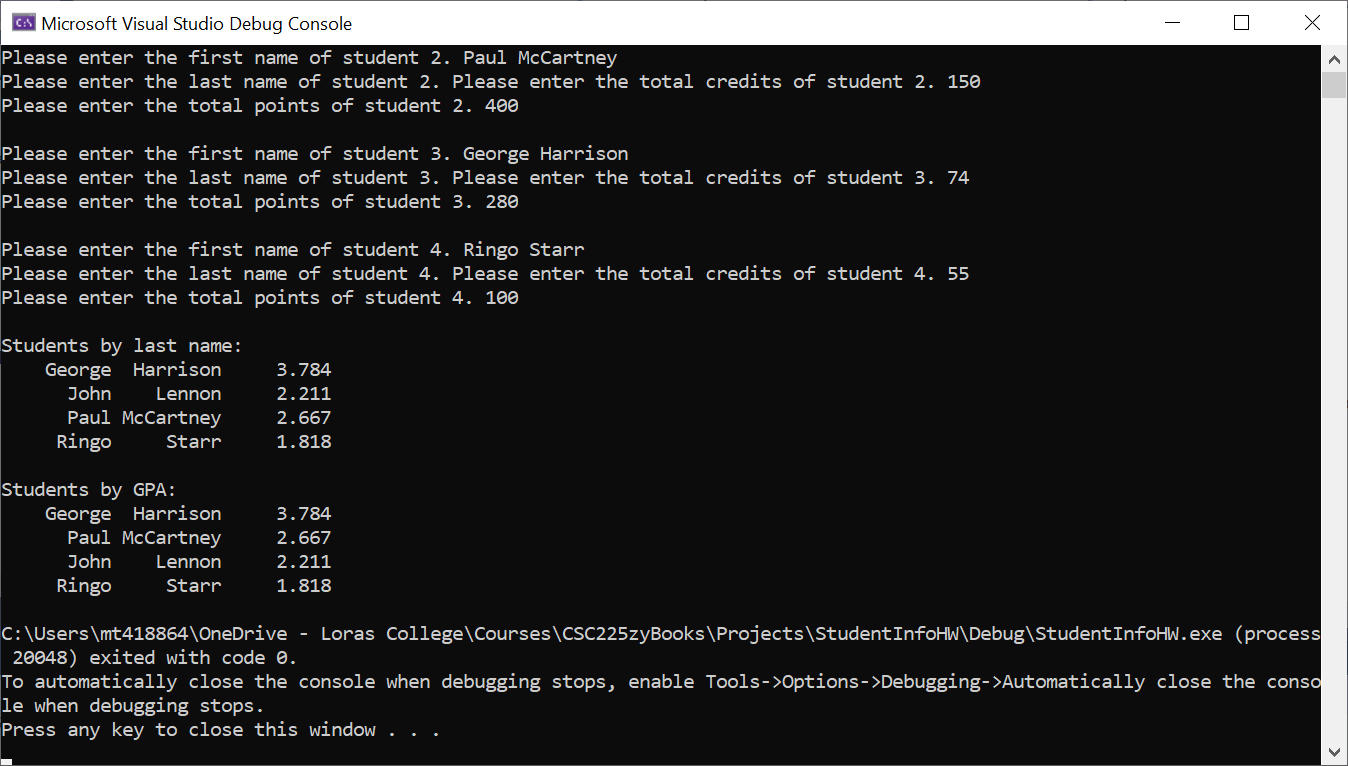
# Instructions

|  |
| --- |
| 1. **You must work on this assignment individually.** You may ask another person to look over your project for errors, but they cannot instruct you the correct way to write. (For example: the person helping can say “The problem is in the ‘switch’ statement” or “Your logic is incorrect in the second ‘if’ statement.”) In addition, you may not use code from other sources, such as the Web or other students. 2. You will need to upload the .cpp and .h files from your solution to eLearn. |

Your task is to write a program that keeps track of the information of up to six students. When the program begins, it should ask the user for the number of students that will be entered and validate that number appropriately. It should then ask for the information about each student, by number, as shown below. All the input for a student should occur in the *enterStudentInfo* function (see below).



After all the students have been added, the program should print the students two ways. First, sorted alphabetically by last name. Then, print the students sorted by GPA from highest to lowest. Both should use standard column widths and show the GPA with three decimal places.



Your *Student* class should follow the specifications in the class slides from September 23.

You must also create the following functions as specified and use them in your program:

|  |  |
| --- | --- |
| **Function Prototype** | **Description** |
| void printStudents(Student[], int) | Is passed the array of students and the number of students. It prints a table of output for the array. |
| void enterStudentInfo(Student[], int) | Is passed the array of students and which student’s information is being read. It asks the user for information about the student and initializes those values for the student. |

You should also have two functions to sort an array of students. On that sorts by last name and one that sorts by GPA.

In addition to the program functioning correctly as described above, you will also be graded on:

* Whether your *Student* class passes all the tests in TestStudent.cpp
* Using appropriate variable names and data types
  + Limit globals
  + Use constants appropriately
  + Name constants using all capital letters
* Using function prototypes
* Using comments appropriately, including header comments in each file
* Formatting your code appropriately using tabs and blank lines
* Having user-friendly and well-formatted input and output
* Utilizing appropriate data validation
  + Number of students
  + Number of credits (more than zero)
  + Total points (you can have from zero to four points per credit)

Additional information:

* Make sure you understand the above requirements before you begin. Dr. Thompson can provide any clarifications you need.