Student Name: ______________________________ *Date:* ________________________

CS221: Data Structures — Programming Assignment #2

Due: At beginning of class on September 14 (15 points)

Problem Statement

You are to develop a program to read Student objects from an input file. You may reuse your student object from program 1. See that assignment for the requirements for a single student object.

New Requirements

You are to implement a **StudentList** data structure/data type that stores players using an internal array of Student in the class. Add Students to the List as they come in from the data file.

You will need to add the class StudentList. The maximum number of students in a list is **25**.

You must implement the following operations on your StudentList along with any other utility functions you might need. You may also need to add operations to the class Student to encapsulate i/o operations to the class.

Operations for StudentList

- Default Constructor
- Add a Student to the List - add items in the order they were found in the file. No sorting is necessary in this version of the program.
- Iterate through the List so that you can get each student out for printing
- getClassAverage
- Clear out the list to make it empty
- Test if a list isFull
- Get the size of the list

New Operations for Student

- Add a method (**read**) that lets a student object read its data from an input stream. NOTE THAT THERE IS NO LONGER a -1 at the end of a set of grades!!! <u>You must determine whether you are at the end of a line or a file.</u>

Example: student1.read(inputfile);

Summary of Operation

- Prompt the user for the input and output file names. DO NOT hardcode file names into your program.
- Open input file
- Read each student and add them to your StudentList
- Keep track of the number of students in the list
- Open an output file
- Write a grade report summary line to the output file, then
- Write each student from the list into the output file, along with any other output required by the assignment
- Do not continue adding students to the list if it is full!

Assumptions

- Grades are no longer marked with a -1 at the end of a set. However, you may assume that there are no BAD grades on the line. They will all be numeric values. It is possible that a line has more than 10 grades, though, so you must deal with it by ignoring the rest of those numbers.
- The student names will be good data. There will not be errors in the data
- Assume all students have 50/50 weights for their average computation.

Other Requirements:

- Please make sure that you create a Win32 Console Application in Visual Studio 2012 or 2015.
- Make sure you create an empty project, with no pre-compiled header.
- Make sure the first comment in your program indicates which version of Visual Studio your project works with.
- Split your program code files into the appropriate .cpp and .h files for each of your classes.
- Name your project **Program2** so I can easily find it on your storage media.

TURN IN:

Please turn in a print out of each of your program files. Submit the electronic version of your project on CANVAS. Make sure your name and the name of the file is in the comments at the top of each program file.

Grading Requirements

- Your program must be well-commented. Comment all variables, functions and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does.
- Use good variable names.
- Use good and consistent naming conventions for class members.
- Use proper code indentation to make sure your program is easy to read and understand.
- You will receive no more than 50% credit if your program does not compile.
- If your program compiles but does not execute correctly, you will receive no more than 70% credit.

Sample Input File and Corresponding Output File

input file

```
Beth Allen
100 100 100
75 75 85 85 95
Joe Davis
50 85 96
50 50 85 85 99
Reza Albedin
88 88 88
45 45 99 99
```

output report

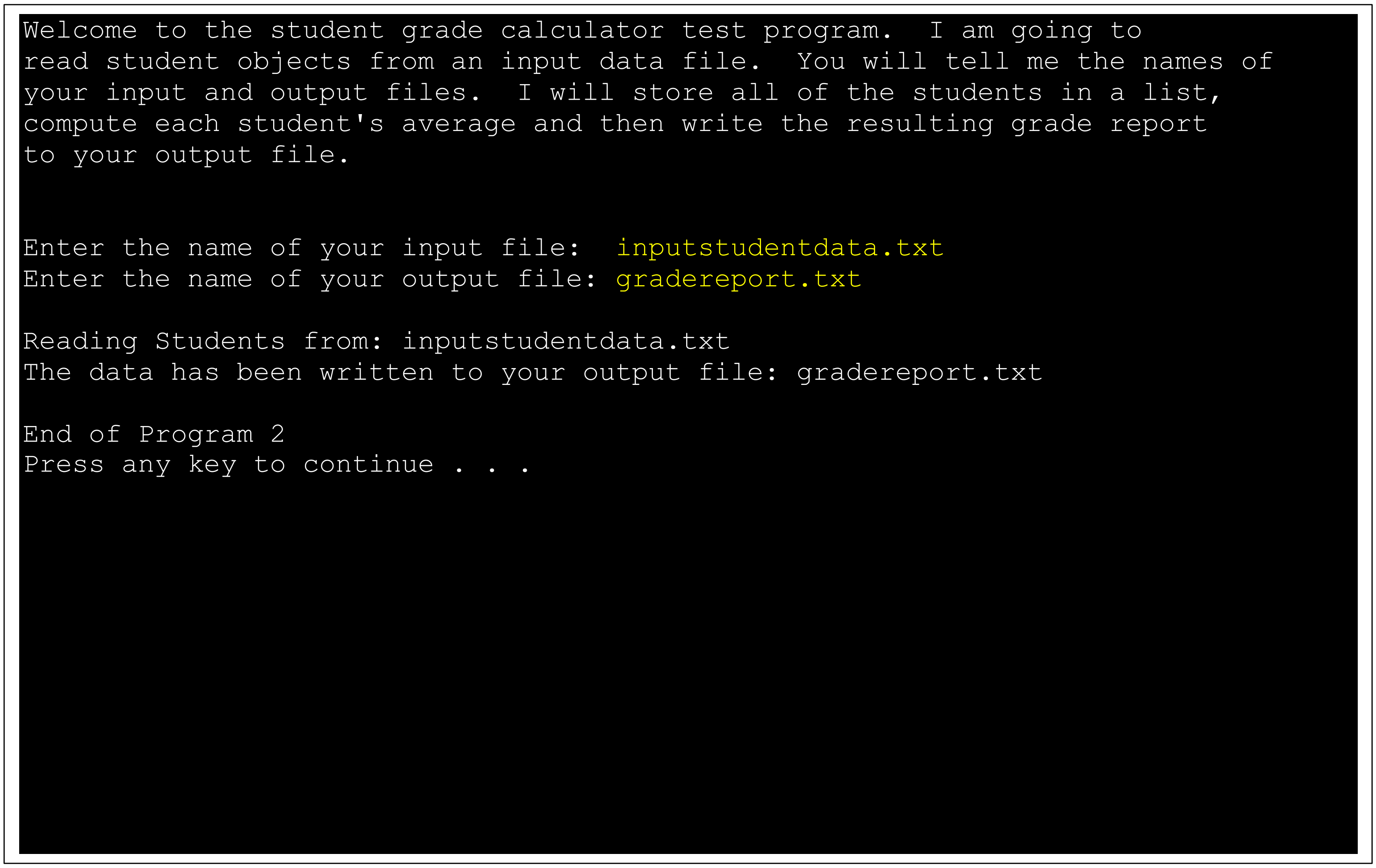
```
COURSE GRADE REPORT ---- 3 STUDENTS ENROLLED
OVERALL CLASS AVERAGE is 82.3

STUDENT NAME   : Allen, Beth
Recorded Grades: 3 tests and 5 homeworks.
Current Average: 91.5

STUDENT NAME   : Davis, Joe
Recorded Grades: 3 tests and 5 homeworks.
Current Average: 75.4

STUDENT NAME   : Albedin, Reza
Recorded Grades: 3 tests and 4 homeworks.
Current Average: 80.0
```

Sample Execution



```
Welcome to the student grade calculator test program.  I am going to
read student objects from an input data file.  You will tell me the names of
your input and output files.  I will store all of the students in a list,
compute each student's average and then write the resulting grade report
to your output file.


Enter the name of your input file:  inputstudentdata.txt
Enter the name of your output file: gradereport.txt

Reading Students from: inputstudentdata.txt
The data has been written to your output file: gradereport.txt

End of Program 2
Press any key to continue . . .
```