Student Name: ______________________________ *Date:* ________________________

CS221: Data Structures Programming Assignment #1

Due: At beginning of class on August 31 (15 points)

Problem Statement

You are to develop a program to further refine our Student class that we started working on day 1 in lecture. You will develop the code for Student, and for a simple main program to test drive an object of type Student.

Your class needs to provide all the methods needed to:

- provide default constructor for Student
  - default first and last name are "unknown"
  - default test and homework weights are 50, 50
- get and set the Student's first name and last name
- add a test grade to the Student
- add a homework grade to the Student
  - (a student will be limited to a maximum of up to 10 test grades and up to 10 homework grades for now)
- Set the student's grade weights (For example weight tests 70% and weight homeworks 30%). Assume that all grades are on a scale of 100 for ease of computing weighted averages.
- Compute the student's average.
- Get (retrieve) the student's average.
- Reset a student object - IE when you invoke reset() on a Student variable it should reset their grade to a cleared state. That means it would look as if there are no tests or homeworks entered yet and that the weights default to values of 50%/50%
- Display a student's basic data to the standard output (see examples later)

You MUST use arrays to store the test grades and the homework grades internal to the Student class. That means that your class will have two arrays as member variables. Do not forget that you need counters internal to the class object to keep track of how many slots in each array are used!!! (I.E. maybe I have a student who has 3 test grade and 5 homework grades.)

TEST DRIVER

Once you have developed a good Student class definition, you need to write a "test" main program to test it out. Your test program should declare a variable of type Student and then prompt the user for the data to be set into the Student. Then, display the characteristics of the Student for the user.

Note that all interactive prompts are coming from the main program. The Student object itself is passive - it is not interfaces with the user by prompting for values. All values are set into the Student by using it's methods (member functions).

The pages at the end of this document have two sample executions that I ran with my test program. Yours should behave and look the same way. (My sample input is highlighted in blue). However, feel free to add a loop to allow the user to repeat the testing, if you wish, rather than just exiting the program (like my sample does) in between each test!

Other Requirements:

- Please make sure that you create a Win32 Console Application in Visual Studio 2012 or 2015.
- Include in the comments at the top of the main program file the version of the compiler you are using. Don't forget your name/date/etc.
- Make sure when you begin, you create an empty project, with no pre-compiled header.
- Name your project **Program1**.
- You may never use built-in C++ template library in this course unless instructed to

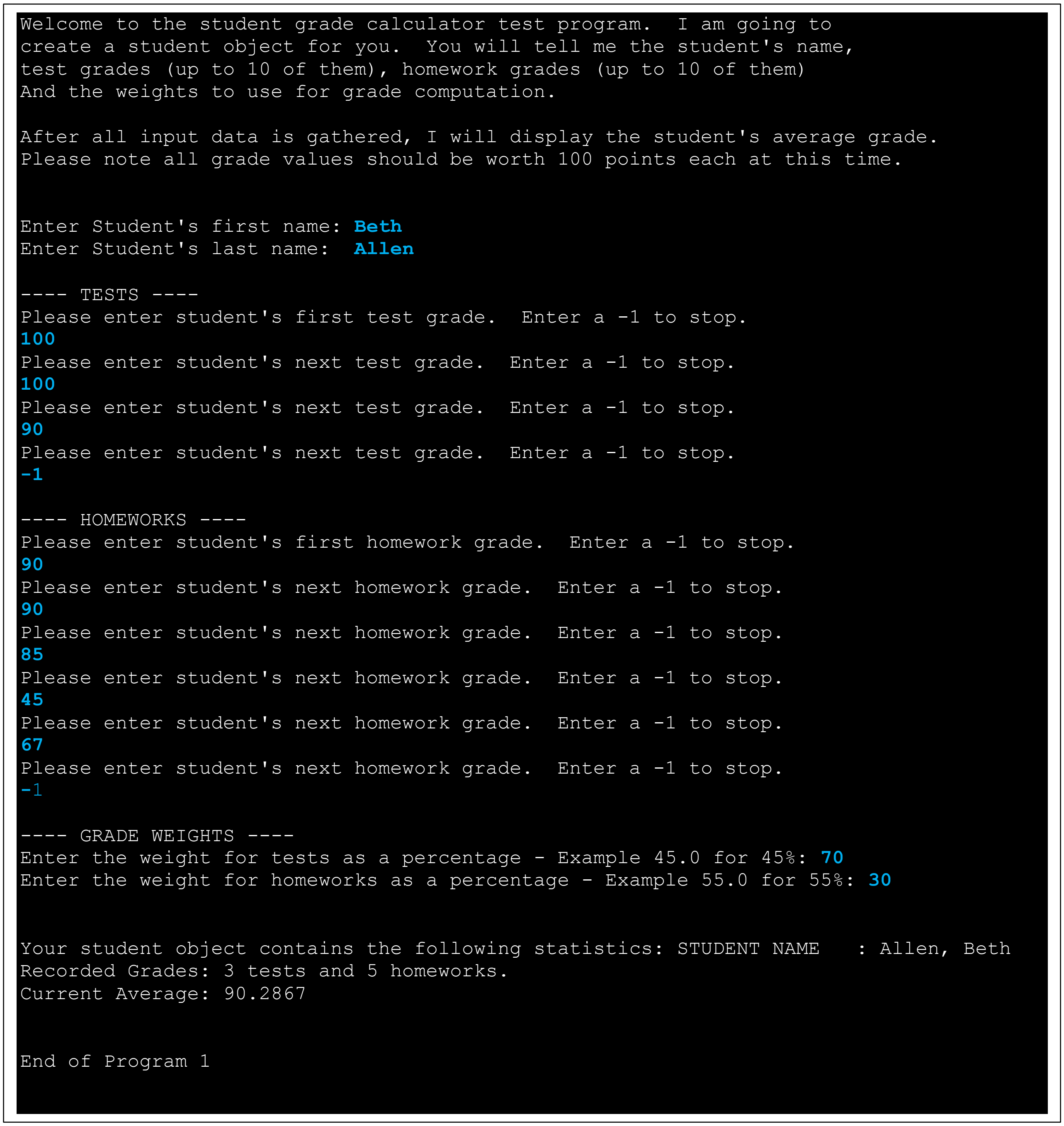
TURN IN:

- Please turn in a print out of each of your program files. DO NOT FOLD your print out.
- Submit the electronic version of your project to canvas for this assignment. Please remember to delete the large database file (extension .sdf) as well as the DEBUG folder from your project. Then zip it up and submit only the single zip file to canvas.

Grading Requirements

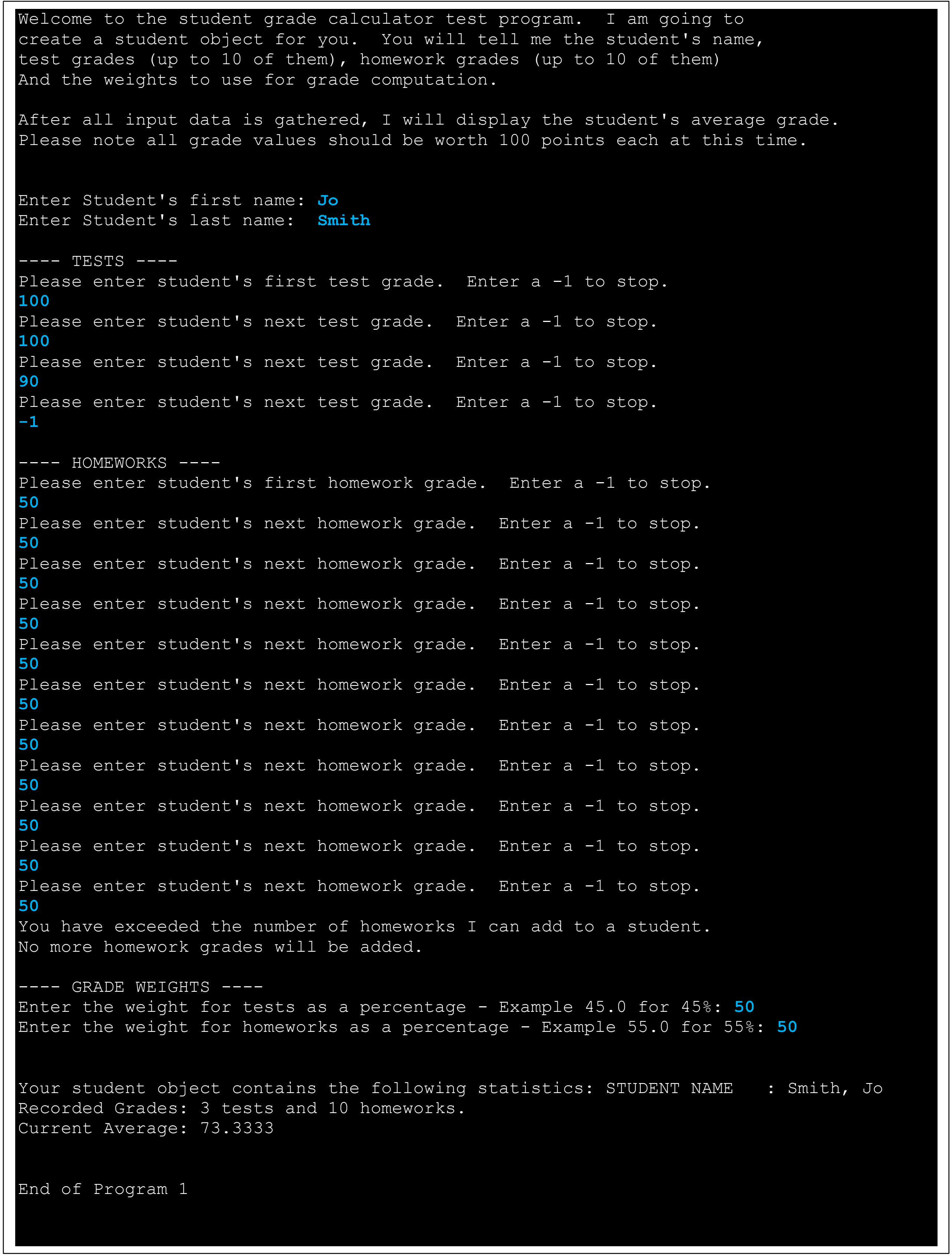
- Your program must be well-commented. Comment all variables, functions and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does. *(Internal documentation on programs in my courses counts for up to 20% of credit.)*
- 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 must include a .cpp file for the main program and a .cpp/.h pair of files for each class that is being implemented in this assignment. Do not put all of the code into a single file. For assignment 1, we are only implementing a single class (Student)
- 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 Execution 1:



```
Welcome to the student grade calculator test program.  I am going to
create a student object for you.  You will tell me the student's name,
test grades (up to 10 of them), homework grades (up to 10 of them)
And the weights to use for grade computation.

After all input data is gathered, I will display the student's average grade.
Please note all grade values should be worth 100 points each at this time.


Enter Student's first name: Beth
Enter Student's last name:  Allen

---- TESTS ----
Please enter student's first test grade.  Enter a -1 to stop.
100
Please enter student's next test grade.  Enter a -1 to stop.
100
Please enter student's next test grade.  Enter a -1 to stop.
90
Please enter student's next test grade.  Enter a -1 to stop.
-1

---- HOMEWORKS ----
Please enter student's first homework grade.  Enter a -1 to stop.
90
Please enter student's next homework grade.  Enter a -1 to stop.
90
Please enter student's next homework grade.  Enter a -1 to stop.
85
Please enter student's next homework grade.  Enter a -1 to stop.
45
Please enter student's next homework grade.  Enter a -1 to stop.
67
Please enter student's next homework grade.  Enter a -1 to stop.
-1

---- GRADE WEIGHTS ----
Enter the weight for tests as a percentage - Example 45.0 for 45%: 70
Enter the weight for homeworks as a percentage - Example 55.0 for 55%: 30


Your student object contains the following statistics: STUDENT NAME   : Allen, Beth
Recorded Grades: 3 tests and 5 homeworks.
Current Average: 90.2867


End of Program 1
```

Sample Execution 2:



```
Welcome to the student grade calculator test program.  I am going to
create a student object for you.  You will tell me the student's name,
test grades (up to 10 of them), homework grades (up to 10 of them)
And the weights to use for grade computation.

After all input data is gathered, I will display the student's average grade.
Please note all grade values should be worth 100 points each at this time.


Enter Student's first name: Jo
Enter Student's last name:  Smith

---- TESTS ----
Please enter student's first test grade.  Enter a -1 to stop.
100
Please enter student's next test grade.  Enter a -1 to stop.
100
Please enter student's next test grade.  Enter a -1 to stop.
90
Please enter student's next test grade.  Enter a -1 to stop.
-1

---- HOMEWORKS ----
Please enter student's first homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
Please enter student's next homework grade.  Enter a -1 to stop.
50
You have exceeded the number of homeworks I can add to a student.
No more homework grades will be added.

---- GRADE WEIGHTS ----
Enter the weight for tests as a percentage - Example 45.0 for 45%: 50
Enter the weight for homeworks as a percentage - Example 55.0 for 55%: 50


Your student object contains the following statistics: STUDENT NAME   : Smith, Jo
Recorded Grades: 3 tests and 10 homeworks.
Current Average: 73.3333


End of Program 1
```