

Assignment 7 - Test Scores

Input Data

For each student, the following data will be input:

- student name (your program should handle names of up to 30 characters - may contain spaces)
- test 1 (an integer value)
- test 2 (an integer value)
- test 3 (an integer value)

Use a struct to hold the name and score for each student, which will be a struct. The name should be stored in a C++ string object.

Validation

Tests are graded on a 100 point scale with a 10 point bonus question. So a valid test score should be 0 through 110.

Processing

Your program should work for any number of students. When the program starts, it should ask the user for the number of students to be processed. Then it should dynamically allocate an array of that size (array of student/score structures) using the **new** operator.

The program should then:

- Call a function to input the student name/score pairs and store them in the array. Input validation: the score must be between 0 and 105, inclusive.
- Call a function to sort the array on the score member in ascending (increasing) order.
- Call a value-returning function that calculates the average of the scores.
- Call a function that displays a neat table of student name/score pairs in sorted order. Include appropriate column heading for your table. Following the table, display the average score with appropriate label.
- Note: you must declare the array parameter using pointer notation instead of array notation, and inside the function you must use pointer notation instead of array notation to access the array elements. If you do not understand this requirement, email me.
- The score is the average of the 3 test scores

Requirements/Hints:

1. Global variables are variables that are declared outside any function. Do not use global variables in your programs. Declare all your variables inside functions
2. Use the C++ string class to hold the student name.
3. Use a struct to store student name/score pairs.
4. Use a dynamically-allocated array of struct to store the information for a class.
5. You must declare the array parameter using pointer notation instead of array notation, and inside the function you must use pointer notation instead of array notation to access the array elements.

Example**Sample Output**

Name	Scores	Letter Grade
Jim Smith	75, 77, 79	C
Victor Montero	81, 89, 90	B
Christa Kim	88, 92, 80	B
<hr/>		
Average	XX.XX	