

---

# Programming Assignment 6: Famous Scientists Diving Competition

COP 3014 - Spring Term 2010

Point Value: 100 points

**Project Due Date: Wednesday April 21, 2010 at 11:59 PM**

**NO programs will be accepted after this due time !!!  
i.e. NO LATE PROGRAMS AND NO EXTENSIONS !!!**

---

## Learning Objectives

- To write a program which requires the creation and manipulation of an *array of structs*
- To utilize the *typedef* construct to create and work with the array of structs
- To utilize C++ strings to work with textual data (the C++ string class)
- To practice more advanced file input
- To make use of a standard sorting algorithm, the *selection sort*
- To perform a common real-world process which is typically done using software

## Problem Statement

In the spirit of the Olympics, we will now work on analyzing scores for an Olympic sport. Twenty-four famous scientists have decided to participate in a special diving competition. The scientists will be judged according to Olympic regulations. There will be two rounds of dives, and each participant will make one dive per round.



Each dive is judged by 7 judges. Points for the dive are assigned according to the following:

1. each judge assigns a score between 0.0 and 10.0 inclusive

2. the highest and lowest scores are eliminated
3. the five remaining scores are summed and the total is multiplied by 0.6. This result is then multiplied by the degree of difficulty of the dive (0.0 to 3.0) to find the total score for the round.

Note: it is possible that more than one score is "lowest," and/or more than one is "highest" - in these cases one of them is eliminated from the summing. In the most extreme case, if all scores were equal, you would eliminate two of them from the "total" calculations.

After the two rounds, the divers with the ten highest totals will be flown to London, England in 2012 to participate in a special diving competition for famous scientists only. Your task is to write a program to evaluate the results of these two initial rounds.

## Input

The data file provided on the class web site is named *diveData.txt*. You may assume the data file is not empty and is completely correct.

You may hardcode the name of the input file directly into your program. You must implement a check to see if the file specified does open, and if it does not, terminate the run using either return in main or exit in another function, as discussed in lectures. If this occurs be sure to print out an appropriate error message for the user.

The data file will contain information for the two rounds of diving competition. First, there will be 48 lines giving the results for each dive in round one, then another 48 lines will give the results for round two. Each dive's data will contain two lines:

1. diver's name, starting in column 1, up to 25 characters
2. dive difficulty level, a real number, then the 7 scores for the dive, all real numbers

The real number values will be separated from each other by at least one blank.

## Output

For this project, you will design the appearance of the output yourself. Use your imagination. Follow the style guidelines for output we have been discussing and demonstrating all semester. Here are the basic requirements for this program:

Print the results of round one and round two in two separate tables. These tables will include this information for each diver: name, degree of difficulty for the dive, the 7 scores, and the total score on this round.

Next, print out a final standings table sorted in descending order by final total score for all divers with this information for each diver: name, final total score (sum of total scores for each round).

Finally, print a list of the *names* of the 10 divers, also in descending order ranked by score, who will be allowed to go on to the next competition.

Note: you must use the *Selection Sort* as presented in lectures to perform sorting tasks in this program.

### EXAMPLE CALCULATIONS

Given the following difficulty rating and scores:

```
Round 1: 2.3      6.0 5.0 6.7 7.8 5.6 8.9 7.6
Round 2: 2.1      6.0 5.0 6.5 7.8 5.6 8.8 5.0
```

```
Round 1 total is: 46.506
Round 2 total is: 38.934
Final total score is: 85.440
```

### Use Of Functions

Part of your grade on this programming project will be determined by how well you utilize functions and parameters appropriately. Start by working on a good design, structure chart, etc. Your program should represent a modular, cohesive design following class style guidelines.

### Use Of Data Structures and Data Types

You must use an *array of structs* as the major data structure on this project with each struct as specified in this write-up. You must use the C++ string class to store and manipulate the diver names.

*You may not use* any user-defined classes on this project. You also *may not use* the C++ Standard Template Library classes or algorithms with the exception of using the C++ string class to store and work with the name string data in this program.

Use the *typedef* construct to create and work with your array of structs type.

You will set up an array of structs which will contain one record for each diver. Each record must contain the diver's name, the degree of difficulty and scores for each of his/her 2 dives, and the total score for the diver. Store the diver's 7 scores on the 2 dives in either a (1) 2-dimensional array field inside the diver's record or (2) two 1-dimensional array fields inside the diver's record.

You should only have one array of structs declared in this program.



To help you get started on this program, here is a set of declarations which you may use for the major data types needed in this program:

```
// one-dimensional array for storing difficulty of dives on each round
typedef double DifficultyList[NUM_ROUNDS];

// two-dimensional array of dive scores
// as an alternative, you may use two separate one-dimensional arrays here
typedef double ScoreTable[NUM_ROUNDS][NUM_SCORES];

// struct to store information for one diver
struct DiverRecord
{
    string name;           // person's name
    double totalScore,    // total score, both rounds
    diveTotal;           // total on current round
    DifficultyList diff;  // difficulty on all dives (note: 1 x 2 array)
    ScoreTable scores;    // scores for all rounds (note: 2 x 7 array)
};

// array of structs of diver records
typedef DiverRecord DiverList[NUM_DIVERS];
```

## Regarding FSU Computer Competency Requirement

This assignment is assessed according to the FSU Computer Competency Requirement. As such students must demonstrate (1) competent use of a discipline-specific software package: Microsoft Visual Studio C++ and (2) the ability to perform simple transactions using the web/Internet: obtaining all of the assignment materials from the class web site and downloading the required file(s), and turning in the completed work correctly to the class web site.

## What File To Turn In and How to Turn In Your Work using Blackboard

Turn in your C++ program source file, which must be named *diving.cpp*

Be 100% certain you have *thoroughly* read the handout entitled "***Submitting Your Program Assignments Electronically Using Blackboard***" for complete instructions on how to submit and how to **verify your submission** after you submit it. This will help you to avoid losing large numbers of project points due to improper submission issues. This handout is available on the course web site under "Handouts."

---

Last Update: March 15, 2010 1:50 PM A. Ford Tyson

---