

User-defined functions**Student grades**

This program requires you to write your own functions to calculate student grades.

Sample input data:

Here is a sample input file, named `scores.txt`, for the program.

Bugs Bunny	77	68	89	99	70	
Donald Duck	45	67	78	89	99	
Mickey Mouse		45	67	89	90	89
Roadrunner	67	55	68	99	97	
Minnie Mouse		77	78	89	90	80

Each input line comprises of a student name followed by the tab character ‘\t’ and five test scores. There is a tab character between each test score too. There might be an indeterminate number of lines in the input file. So you should design your program to read until input failure.

Output

You are supposed to print out the name of the student, the five test scores, the average score and a letter grade that is dependent on the average score. The letter grade is calculated based on the following ranges: A for 90.00 – 100.00, B for 80.00 – 89.99, C for 70.00 – 79.99, D for 60.00 – 69.99 and F for 0.00 – 59.99.

Sample output:

Here is a sample output file, named `grades.txt`, which corresponds to the input data given above:

```

Programmer: Mir Farooq Ali
CS 1044 Spring 2004 Project 6-b

=====
      Name          Test1  Test2  Test3  Test4  Test5  Average Grade
=====
Bugs Bunny          77     68     89     99     70     80.60     B
Donald Duck         45     67     78     89     99     75.60     C
Mickey Mouse        45     67     89     90     89     76.00     C
Roadrunner          67     55     68     99     97     77.20     C
Minnie Mouse        77     78     89     90     80     82.80     B
=====

```

The first two lines identify the programmer and project, as usual. Then there is a blank line, followed by three header lines. This is followed by the name, their five test scores, test average and a letter grade for each student. Finally, a trailing line is printed out.

If you have read the description of how the Curator scores your program in the *Student Guide*, you know that is important that you use the same spelling and capitalization for all the labels shown above. The horizontal spacing does not effect scoring unless you combine things that should be separate or separate things that should be combined. You are free to experiment with the horizontal layout but you should try to align the columns neatly.

Note that you must insert blank lines and divider lines as shown.

User-defined function details

You are supposed to implement three functions in your program. The three function definitions are given below:

```
1. int sumScores(int test1, int test2, int test3, int test4, int test5);
```

```
// This function should return the sum of the five test scores

2. char calculateLetterGrade(double average);
   // This function calculates the letter grade based on the ranges indicated
   // earlier

3. void findAverage(int sum, double& average);
   // This function finds and returns the average test score using the
   // pass-by-reference mechanism
```

Evaluation:

Everything that was said in the specification for the earlier about testing still applies here. Do not waste submissions to the Curator in testing your program! There is no point in submitting your program until you have verified that it produces correct results on the sample data files that are provided. If you waste all of your submissions because you have not tested your program adequately then you will receive a low score on this assignment. You will not be given extra submissions.

Your submitted program will be assigned a score, out of 100, based upon the runtime testing performed by the Curator System. We will also be evaluating your submission of this program for documentation style and a few good coding practices. This will result in a deduction (ideally zero) that will be applied to your score from the Curator to yield your final score for this project.

Read the *Programming Standards* page on the CS 1044 website for general guidelines. You should comment your code in the same manner as the code given for the first two programming assignments. In particular:

- You should have a header comment identifying yourself, and describing what the program does.
- Every constant and variable you declare should have a comment explaining its logical significance in the program.
- Every major block of code should have a comment describing its purpose.
- Adopt a consistent indentation style and stick to it.

Your implementation must also meet the following requirements:

- Choose descriptive identifiers when you declare a variable or constant. Avoid choosing identifiers that are entirely lower-case.
- Use named constants instead of literal constants when the constant has logical significance.
- Use C++ streams for input and output, not C-style constructs.
- Use C++ string variables to hold character data, not C-style character pointers or arrays.

Understand that the list of requirements here is not a complete repetition of the *Programming Standards* page on the course website. It is possible that requirements listed there will be applied, even if they are not listed here.

Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* can be found at: <http://ei.cs.vt.edu/~eags/Curator.html>

The submission client can be found at: <http://eags.cs.vt.edu:8080/curator/>

Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
//   anyone other than my instructor or the teaching assistants  
//   assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
//   or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
//   was obtained from another source, such as a text book or course  
//   notes, that has been clearly noted with a proper citation in  
//   the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
//   interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.