

**HW #5****Complete By: Monday Feb 17<sup>th</sup> @ 9:00pm****Policy: Individual work only, late work not accepted****Submission: electronic via Blackboard (see below)****Your Assignment**

Your assignment is to write a grading program. Assume the existence of four input files, residing in the same directory as your F# program: "exam1.txt", "exam2.txt", "exam3.txt", and "final.txt". Assume that all four files contain the same number of scores  $N$ , where  $N > 0$ . Each score appears on a line by itself, and may be a real number. For example, the "exam1.txt" input file might look like this:

```
72
88.5
65
```

The first line of each file represents the scores for student 1, the second line of each file represents the scores for student 2, and so on. For example, suppose the four inputs files are as follows (the left-most column represents "exam1.txt", the 2<sup>nd</sup> column "exam2.txt", the 3<sup>rd</sup> column "exam3.txt", and the right-most column "final.txt"):

<b>90</b>	<b>90</b>	<b>80</b>	<b>100.0</b>
<b>100</b>	<b>22</b>	<b>100</b>	<b>60.0</b>
<b>40</b>	<b>70</b>	<b>50</b>	<b>90.0</b>

This implies that student 1 scored 90, 90, and 80 on the exams, and 100 on the final. To compute a student's final score, you drop the lowest exam score (the final is never dropped), and then weigh the remaining scores as follows: first exam score 30%, second exam score 30%, and final 40%. For example, in the case of student 1, we drop exam #3 (80), yielding a final score of  $(90*0.3) + (90*0.3) + (100*0.4) = 94.0$ .

Write an F# program that inputs the data from the four input files, and outputs the final score for each student to the console. For the input files above, your output should be

```
** Final Scores **
```

```
94.0
```

```
84.0
```

```
72.0
```

In addition, your program must also output the following statistics based on the final scores: the maximum, the minimum, the average, and the standard deviation<sup>1</sup>. Continuing our example, this implies your program

<sup>1</sup> There's a nice definition with example in Wikipedia: [http://en.wikipedia.org/wiki/Standard\\_deviation](http://en.wikipedia.org/wiki/Standard_deviation)

should then output:

```
** Statistics **  
Maximum: 94.0  
Minimum: 72.0  
Average: 83.33333333  
Standard deviation: 8.993825042
```

Your output may differ slightly — the number of digits to the right of the decimal point may change based on your version of Visual Studio / F# / Windows (it shouldn't, but it's okay if it does).

In writing your program, use a purely-functional style (no side-effects), with an exception for any functions performing I/O — I/O is inherently full of side-effects in the operating system. And you are encouraged to use the higher-order functions like **map** and **reduce**. However, the idea of the assignment is to learn functional programming, so do not use functions that trivialize the assignment. For example, .NET and F# contain functions that will compute

- Min
- Max
- Sum
- Average
- Length
- Sort
- Reverse
- and so on...

Do not use these function, or functions like them — using these functions will trivialize the assignment. Instead, the requirement is that you must write your own functions, from scratch, for each of the components of this exercise: min, max, sum, length, average, standard deviation, and so on<sup>2</sup>. You can write them recursively, or using higher-order functions. When in doubt, ask on **Piazza**.

NOTE: you need to print the exam scores on separate lines. It's not hard to do, but you run into a strange warning because `printfn` is a function that returns the type "unit", which you can't really do anything with — yet you need to tell F# to "ignore" it. First, here's a function to print the elements of a list on separate lines:

```
let printfnList(L) =  
    printfn "*** Elements of List ***"  
    List.map (printfn "%A") L
```

When it comes time to call this function, normally you would do this:

```
printfnList( [1; 2; 3] )
```

---

<sup>2</sup> Feel free to use any code presented in class.

But you'll get a strange warning about needing to ignore the returned "unit list", so the solution in F# is to use the ignore( ) function in the call:

```
ignore( printfnList(...) )
```

Strange but true :-)

## Visual Studio

Now that we are working with input files, <http://ideone.com> is no longer a viable option. Instead, you are required to work with Visual Studio 2012 or 2013, which runs only under Windows. Feel free to continue to use <http://ideone.com> as a convenient test environment, but your final program must be written using Visual Studio.

## Electronic Submission

Using Blackboard ( <http://uic.blackboard.com/> ), submit your F# Source File (.fs) under the assignment "HW5". Please make sure your submission contains a header comment with your name, and the following:

```
//  
// F# grading program.  
//  
// <<YOUR NAME HERE>>  
// U. of Illinois, Chicago  
// CS341, Spring 2014  
// Homework 5  
//
```

You may submit as many times as you want before the due date, but we grade the last version submitted.

## Policy

Late work is not accepted. All work is to be done individually — group work is not allowed. Academic dishonesty is unacceptable, and all parties involved will be immediately subject to the official academic integrity review process. The University's policy is quite clear, and can be read here: <http://www.uic.edu/depts/dos/studentconduct.html>. In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Examples of academic dishonesty include emailing your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you.