

Lab 5

Due Feb 27 by 11:59pm **Points** 50 **Submitting** a file upload **File Types** cpp

Statistics With Functions and Arrays: Chapters 6 and 7

The [General Programming Lab Instructions](#) apply to this assignment.

Please note the following:

- You have two weeks to complete this lab because it is larger and more challenging than previous labs. Please start this lab as soon as possible.
- One of the main purposes of this assignment is to give you experience passing arrays to and returning them from functions. All four functions listed in this assignment are required.

Statistics are valuable tools that are used in a wide variety of disciplines, including computer science. Your program will calculate two simple statistics on data (a list of scores) entered by the user: a histogram and a standard deviation. (To calculate the standard deviation, your program will also need to calculate the mean or average.)

To help with the data input, you may find it useful to review

- [average.cpp \(version 3\)](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html>), [chi2.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html>) and
- the chapter 6 [Practice Problems: Functions With Loops](http://icarus.cs.weber.edu/~dab/cs1410/textbook/6.Functions/practice/loops.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/6.Functions/practice/loops.html>) 6.13.1 (2, 3, 5, and 6)

To help convert the Σ operator used in the formulas into C++, see the [chi2.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html>) and the [rms.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/rms.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/rms.html>) examples.

Histograms

Histograms (<https://en.wikipedia.org/wiki/Histogram>) are a way of graphically summarizing and presenting data. They summarize data by grouping events, values, or occurrences into ranges or bins (e.g., all all of the red stuffed toys go into one bin, all the blue toys in another bin, etc.). Histograms typically consist of a set of vertical rectangles whose height corresponds to the number of occurrences within a range or bin (if there are six red toys and three blue toys, then the red rectangle will be twice as tall as the blue rectangle). To make our programming task easier, our histograms will consist of '*' characters drawn horizontally (see the test cases below).

Each element in the bins array is a counter that counts how many scores were in a specific range. So, printing the histogram is *similar* to printing the pyramid of Xs or drawing the pine tree (especially drawing the tree trunk). Use two for loops, one nested inside the other. The outer loop walks through the bins array and the inner loop prints each '*' on a given line.

Standard Deviation

There are two standard deviations in common use:

1. Population: The standard deviation based on the entire population (i.e., it treats the data as the entire population)
2. Sample: An estimate of the standard deviation based on a sample (i.e., it treats the data as a sample of the the population)

The formula for the population standard deviation (which is what we will use for lab 5) is: $SD = \sqrt{\frac{\sum (X_i - M)^2}{N}}$

X_i

X is an array of scores, so X_i is score_i (i.e., score[i] - the score at index location i in the scores array)

M

Mean (i.e., the average of all scores) - see [average.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html)

(<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html>)

N

The number of scores (i.e., the count)

Also, the Σ symbol is an operator that just means to "sum" an array of values: subtract the mean from each score, square the result, and sum all of the squares; divide the sum by N and take the square root. [chi2.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/chi2.html>) and [rms.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/rms.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/rms.html>) demonstrate how to program formulas that contain the Σ operator.

Program Requirements

1. Name your program file `stats.cpp`
2. Do not use any **global variables** (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/6.Functions/scope.html>).
3. Use only `iostream` and `iomanip` functions for I/O and formatting (no `stdio`) - but may use other header files such as `<cmath>`
4. Your program will have four functions:
 - a. `main`
 - b. `mean`
 - c. `deviation`
 - d. `histogram`
5. `main`
 - a. Define a variable to maintain a count of the number of scores entered and initialize it to 0
 - b. Define the scores array (the scores array does not need to be initialized)
 - c. Allow for a maximum (not total) of 100 scores (see [Arrays and Loops](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/arrays_loops.html) (http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/arrays_loops.html) - the examples at the bottom of the page)
 - d. Read a list of integers from the console one at a time and place each integer into the scores array
 - e. None of the entered scores will be < 0
 - f. Increment count for each score read and stored in the scores array
 - g. Stop the data entry and begin the calculations when the user enters a -1 (do not include the -1 in the data)
 - h. Call the `histogram` function (pass any arguments that are needed)
 - i. Print the histogram (the histogram function fills and returns the bins array but you must print the histogram in `main` - see the "Test Cases" below for an example of the required output); this sub-problem is similar to drawing the trunk of the pine tree
 - j. Call the `deviation` function, which returns the standard deviation
 - k. Print the standard deviation (steps j and k may be combined into a single statement if desired)
6. `mean`

- a. Pass in whatever arguments are needed - see [average.cpp](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/progexample/average.html>)
 - b. Return a double value that is the mean or average of the scores
7. deviation
- a. Pass in whatever arguments you need
 - b. Return a double that is the standard deviation of the scores
8. histogram (the histogram function fills and returns the bins array - it does NOT print the histogram)
- a. Calculate the histogram (i.e., fill but do not print the bins array)
 - b. Returning an array from a function presents a special problem. The text presents the problem and three possible solutions. Choose and implement one of the three techniques illustrated in [Arrays And Functions](http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/arrays_funcs.html#return) (http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/arrays_funcs.html#return).
 - c. Each element in the bins array is really an accumulator and so must be initialized to 0. See [Figure 2](#) (<http://icarus.cs.weber.edu/~dab/cs1410/textbook/7.Arrays/initialize.html>) for an example.
 - d. Group the scores in the histogram as follows
- ```
Bin 9: score ≥ 90
Bin 8: score ≥ 80 but < 90
Bin 7: score ≥ 70 but < 80
 .
 .
 .
Bin 1: score ≥ 10 but < 20
Bin 0: score < 10
```
- e. Note that some scores in the input include extra credit (i.e., are > 100); your program must work for scores up to 109.

## Test Cases

**Case 1:** 100, 95, 90, 85, 80, 75, 70, 65, 60, 40, 20, and 5.

```
9| ***
8| **
7| **
6| **
5|
```

```
4| *
3|
2| *
1|
0| *
SD: 28.5379
```

**Case 2:** 30, 40, 45, 102, 35, 42, 65, 89, 55, 48, 56, 46, 42, 54, 56, 51, 47, 50, 51, 50, 50, 47, 52, 53, 47, 44, 69, 35, 40, 45, 35, 42, 65, 55, 48, 100, 56, 46, 42, 54, 56, 51, 47, 50, 51, 50, 50, 47, 52, 53, 47, 78, 80, and 95

```
9| ***
8| **
7| *
6| ***
5| *****
4| *****
3| ****
2|
1|
0|
SD: 15.2579
```

## Lab 5 Scoring

| Criteria                                                                                                                                                                                                                                                                                                                                                                                      | Ratings | Pts      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------|
| <b>File Name</b><br>Program file is named correctly (stats.cpp). It's okay if Canvas adds a -1, -2, etc. to the file name.                                                                                                                                                                                                                                                                    |         | 1.0 pts  |
| <b>Data Input</b><br>Values stored in scores and count are correct. Input ends when the user enters -1; the -1 is not included with the data. No unspecified prompts (e.g., don't prompt for the number of scores, or ask yes/no or true/false questions).                                                                                                                                    |         | 5.0 pts  |
| <b>Function Calls</b><br>The function called from main and any needed local variables are correct.                                                                                                                                                                                                                                                                                            |         | 5.0 pts  |
| <b>Print histogram</b><br>Histogram is printed correctly as illustrated.                                                                                                                                                                                                                                                                                                                      |         | 8.0 pts  |
| <b>mean Function</b><br>The mean or average is correct. May be called in main or in deviation.                                                                                                                                                                                                                                                                                                |         | 3.0 pts  |
| <b>deviation Function</b><br>The value of the standard deviation is correct and printed in main.                                                                                                                                                                                                                                                                                              |         | 5.0 pts  |
| <b>histogram Function (1)</b><br>The counts in the bins array are correct.                                                                                                                                                                                                                                                                                                                    |         | 8.0 pts  |
| <b>histogram Function (2)</b><br>The counts in the bins array is correctly returned to main, where it is printed, using one of the three outlined solutions.                                                                                                                                                                                                                                  |         | 10.0 pts |
| <b>Miscellaneous</b><br>Program does not use global variables and does not use stdio.h functions. The program is structured correctly as demonstrated in the examples. The program is clean and easy to read: indentation is consistent, blank lines are used appropriately; the program is created "Empty," and all "pause" statements and/or dummy reads are commented out or removed, etc. |         | 5.0 pts  |

| Criteria                                                                                                                                                                                                                                                              | Ratings | Pts     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|
| Extra Credit 1<br>In main: make the code more compact by printing the histogram using only one loop - no if or conditional statements, no calls to functions you write. Hint: see the posted solutions for lab 3.                                                     |         | 2.0 pts |
| Extra Credit 2<br>In histogram: make the code more compact by filling the bins array using only computation (a) only one loop, (b) no function calls, (c) no if-statements or conditional statements, (d) no moving logic to the main where the histogram is printed. |         | 3.0 pts |
| Total Points: 55.0                                                                                                                                                                                                                                                    |         |         |