

## COSC 1410, Spring 2016

### Assignment 6

**[1] Objective:** This assignment is designed to test your understanding of functions and arrays. More specifically we are looking for: function definitions and prototypes, call-by-value and call-by-reference parameters, and also array definition and usage.

**[2] Description:** In this assignment, you will generate a random array used to represent the grades in a class and then you will calculate simple statistical methods used to analyze a data samples and curve the grades.

Here is what you need to do.

- Generate an array of 100 integer scores (0-100).
- Compute the mean, min, and max. (What we really need is the median which in most cases is close to the mean.)
- Calculate the total number of each letter grades using the standard grade scale.
- Print the results of all the computations.
- Curve the grades.
- Compute the mean, min, and max of the curved grades.
- Calculate the updates total number of each letter grade using the standard grade scale.
- Print the updated result of all the computations

Percentage	Letter Grade
90-100	A
80-89	B
70-79	C
60-69	D
<59	F

Requirements:

1. You should generate one array called score where you will store 100 randomly generate scores .
2. You should not use global variables except for constants.
3. You must write the following functions. These functions must use some call-by-value parameters and some call-by-reference parameters. I will leave them to you to decide on what parameters are needed for each function.
4. To curve the exam, use the following algorithm. If the score  $x$  is  $(x - \text{mean}) / (\text{maxGrade} - \text{mean}) * 25 + 75$ .

Suggested functions:

- `getScore()`: read one randomly generated score in a range
- `getScores()`: fill up all 100 numbers
- `getStat()`: get mean, stdev, min, max
- `computeLetterGrade()`: compute the letter grades
- `curveGrades()`: curves the grades

**[3] Input:** Since we don't have a convenient way to get a large number of scores into the program, we have decided to generate the scores by using the random number generator. For the scores to look somewhat real, this is what you need to do. (1) Generate 20% integer numbers between 0 and 100, (2) Generate 30% integer numbers between 40 and 90, (3) Generate the other 50% of the numbers between 60 and 80 for a total of 100 numbers, (4) Set N = 100 (You may want to test your program with a smaller N first). Your program should save all these numbers in an array.

**[4] Output:** A sample output looks like:

s[0] = 24 s[1] = 23 s[2] = 8 s[3] = 19 s[4] = 21 s[5] = 82 s[6] = 29 s[7] = 90 s[8] = 94 ... s[96] = 70 s[97] = 84 s[98] = 76 s[99] = 63  Mean is 66.11 Standard Deviation is 21.79 Lowest Grade is 0 Highest grade is 95  There were a total of 3 A's, 29 B's, 22 C's, 23 D's, and 23 F's.  24 => 38 23 => 37 8 => 24 19 => 34 21 => 35 82 => 88 29 => 42 90 => 95	94 => 99 7 => 23 ... 11 => 27 25 => 39 0 => 17 95 => 100 4 => 21 ... 87 => 93 80 => 87 80 => 87 69 => 77 66 => 74 71 => 79 64 => 73 70 => 78 84 => 90 76 => 83 63 => 72  After the curve:  Mean is 74.51 Standard Deviation is 18.88 Lowest Grade is 17 Highest grade is 100  There were a total of 18 A's, 28 B's, 29 C's, 12 D's, and 13 F's,
---	--

**[5] Due date:** Wednesday, **March 23, 2016.**