CS 5000 – Summer 2025
Assignment #11, 50 points

Recursion – Chapter 18

Please remember to read and apply coding practices outlined in assignment 8.

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as shown below. ***Please use only concepts and programming constructs/syntax we discuss to date.*** Make sure you **include a header for each program (see previous assignments).**

Program #1 (25 points): Write a java program, named ***Vowels***, as follows: The main method prompts the user to enter a string. The main method then passes the input string to method `countVowels(…)` to **recursively** count and then return number of vowels (*a, e, i, o, u*) in the passed string. Again, method `countVowels(…)` is a recursive method. Format the outputs as follows. Test data shows values for illustration, user may enter values one per line. Test data does not show input prompts. Make sure your code displays the outputs following the test data format. Non-recursive solutions receive no points.

Test data:

```
Entered string:    This is a test input
Number of vowels:  6

Entered string:    Hello world
Number of vowels:  3

Entered string:    My list of Objects
Number of vowels:  4
```

Document your code, use proper prompts for input, format outputs as shown above, use sound coding practices we learned thus far, do not hard code inputs, allow program re-runs, and test your code thoroughly.

Program #2 (25 points): Write a java program, named ***countAverageGrades***, as follows: The main method prompts the user to enter number of students in a class (class size is integer value), then prompts the user to enter the grades (between 0 and 100) into an array of type integer. The entered class size determines the array size. Use exception handling to verify that entered grades are between 0 and 100, reject invalid inputs. Next, the main method passes the filled array to method `findAverage(…)` to **recursively** determine and return the class average as a double value. Again, method `findAverage(…)` is a **recursive** method. This may take another parameter if needed. Format the outputs as follows. Shown input values are just for illustration, user may enter values one per line. Test data does not show input prompts, only outputs. Make sure your code displays the outputs following the test data format. Non-recursive solutions receive no points.

Test data:

```
Class size:        3
Entered grades:    100 100 100
Class average:     100.00
```

```
Class size:        7
Entered grades:    50 75 80 80 40 35 85
Class average:     63.57

Class size:        8
Entered grades:    0 100 25 90 55 30 90 35
Class average:     53.13
```

Document your code, use proper prompts for input, format outputs as shown above, use sound coding practices we learned thus far, do not hard code inputs, allow program re-runs, and test your code thoroughly.

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines posted in D2L. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment due date is posted in D2L.
3. Please compile, run, and test your code right before you upload your java files to the assignment submission folder in D2L.
4. Please upload only the .java files (total 2 files).