**Contestant Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Rank: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**C++ PROGRAMMING**

(335)

**REGIONAL 2024**

**APPLICATION KNOWLEDGE:**

Project/Solution Files (30 points)

Job1 (260 points)

Job2 (140 points)

**TOTAL POINTS (430 points)**

**Test Time: 90 minutes**

**GENERAL GUIDELINES:**

*Failure to adhere to any of the following rules will result in disqualification:*

1. Contestant must hand in this test booklet and all printouts if any. Failure to do so will result in disqualification.
2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests (handwritten, photocopied, or keyed) are allowed in the testing area.
3. Electronic devices will be monitored according to ACT standards.

Your application will be graded on the following criteria:

Job 1

You must create a C++ console application named CPP\_335\_Job1\_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, CPP\_335\_Job1\_01\_2345\_6789.

Your contestant number must appear as a comment at the top of the main source code file.

Add a C++ program to the project named Job1.cpp that reads a story from a provided text file name “Story.txt”. This text file is not to be edited. Your program will need at least four function/methods that take in the text from the story. All four functions will need to take in the text as a parameter. No global should be used. If there are any problems opening the file, the program should print “Error Opening File” and terminate the program.

The first function will calculate the number of spaces that are present in the story and return the number as an integer.

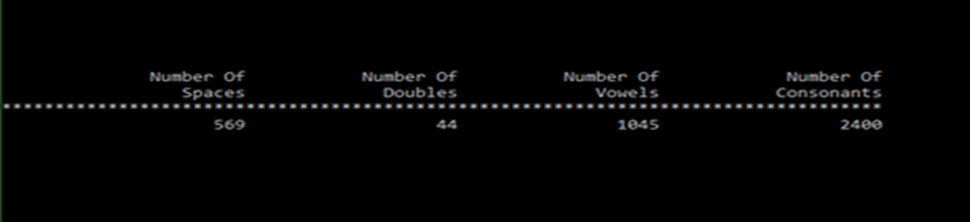
The second function will calculate the number of duplicate letters side-by-side. In example, the word “apple” has on letter that is duplicated side-by-side. The function will return the calculated number as an integer.

Based on the English language, the third function will calculate the number of vowels used in the story. The function will return the number as an integer.

Based on the English language, the fourth function will calculate the number of consonants used in the story. The function will return the number as an integer.

The output should closely resemble Figure 1 below. Do not use spaces or tabs to space the calculated values.

All functions/methods and any code segments need to be commented thoroughly.



*Figure 1*

Job 2

You must create a C++ console application named CPP\_335\_Job2\_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, CPP\_335\_Job2\_01\_2345\_6789.

Your contestant number must appear as a comment at the top of the main source code file.

Write a program that can read multiple integers from the text file named “grades.txt” and store them in a data structure of your choice. The text file is provided and is not to be edited. The program needs to display “Error reading/opening file” if there is a problem opening the datafile.

Your program will then need to have two functions, one named getMean and the other will be named getStdDeviation. There should be no global variables in your project.

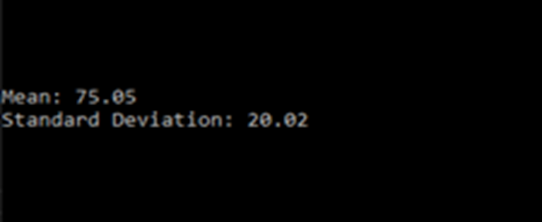
The function getMean will take in the list as of numbers and calculate and return the calculated value as a double.

The function getStdDeviation will need to take in two parameters. The first will be the list of numbers and the second will be the average/mean of the number list. The program should solve for the standard deviation by using the following the steps below:

1. Keeping a running sum of the following: each number in the list subtracted from the average value then squared.
2. Divide the running sum by the amount of the numbers in the list. For example, if there are 10 numbers in the list, then the running total is divided by 10.

The value for the standard deviation will be returned from the function as a double.

Your program will then print both values out to the screen using the data labels of “Mean: “ and “Standard Deviation: “. See Figure 2



*Figure 2*

Your application will be graded on the following criteria:

**Solution and Project**

The projects are uploaded or present on the flash drive \_\_\_\_ 10 pts

The projects are named according to the naming conventions \_\_\_\_ 10 pts

**Program Execution**

Code for both projects are copied to USB/upload drive and both   
programs run from USB/upload drive \_\_\_\_ 10 pts

Solution and Project Files 30 pts

*If the program does not execute, then the remaining items in this section receive a score of zero.*

**Job 1**

Application displays error message if problem opening the file \_\_\_\_ 10 pts

Application displays the correct number of spaces \_\_\_\_ 10 pts

Application displays the correct number of double letters \_\_\_\_ 10 pts

Application displays the correct number of vowels \_\_\_\_ 10 pts

Application displays the correct number of consonants \_\_\_\_ 10 pts

The output is formatted correctly \_\_\_\_ 10 pts

**Source Code Review**

Code is commented at the top, for each function, and code blocks as needed \_\_\_\_ 10 pts

Code uses reasonable and consistent variable naming conventions \_\_\_\_ 10 pts

Data types are handled in a logical and consistent manner (no global variables) \_\_\_\_ 20 pts

countSpaces function is designed and used correctly \_\_\_\_ 30 pts

countDoubles function is designed and used correctly \_\_\_\_ 30 pts

countVowels function is designed and used correctly \_\_\_\_ 30 pts

countConsonants function is designed and used correctly \_\_\_\_ 30 pts

Output is coded correctly to maintain the four columns \_\_\_\_ 40 pts

**Job1 Points**: 260 pts

**Job 2**

Application displays error message if problem opening the file \_\_\_\_ 10 pts

Application displays the correct answer for mean \_\_\_\_ 10 pts

Application displays the correct answer for standard deviation \_\_\_\_ 10 pts

The output is formatted correctly and shown with only 2 decimal places \_\_\_\_ 10 pts

**Source Code Review**

Code is commented at the top, for each function, and code blocks as needed \_\_\_\_ 10 pts

Code uses reasonable and consistent variable naming conventions \_\_\_\_ 10 pts

Data types are handled in a logical and consistent manner (no global variables) \_\_\_\_ 20 pts

getMean function is designed and used correctly \_\_\_\_ 30 pts

getStdDeviation function is designed and used correctly \_\_\_\_ 30 pts

**Job1 Points**: 140 pts