**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 14 – Omar Faruk**

**25 points – Due November 2, 11pm**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Paste your code and screenshots into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

You've been hired by *Text Turtles* to write a C++ console application that determines the reading grade level of the text in a file. The text comes from two files: SampleText1.txt and SampleText2.txt. Place the files where your app can find them. File SampleText1 will be used to test your app. File SampleText2 will be used for your final run.

In your app, attempt to open the input file. Print an error message if the file doesn't open. Otherwise, read one line at a time from the file and keep count of the number of lines read. For each line read, loop through the line and count the number of:

● Alphanumeric characters – use function **isalnum**.

● Punctuation characters – use function **ispunct**.

● Spaces.

● Other characters – this should be zero.

● Number of sentences – this is the number of periods (.).

● Number of characters.

After reading the file, these counts will be for the entire file. Then calculate the following:

● Number of words – this is the number of spaces plus the number of sentences.

● Number of letters in words – this is the number of alphanumeric characters.

● Factor1 = number of letters in words / words \* 100

● Factor2 = sentences / words \* 100

● Reading level = (0.0588 \* Factor1) – (0.296 \* Factor2) – 15.8

Use formatted output manipulators (setw, left/right) to print the following rows:

● Alphanumeric characters.

● Punctuation characters.

● Spaces.

● Other characters.

● Number of characters.

● Number of sentences.

● Number of words.

● Number of letters in words.

● Factor 1.

● Factor 2.

● Reading level.

And two columns:

● A left-justified label.

● A right-justified value.

Define constants for the file names and column widths. Format all real numbers to one decimal place. For file SampleText1, the output should look like this:

Welcome to Text Turtles

-----------------------

Reading lines from file 'SampleText1.txt' ...

12 line(s) read from file 'SampleText1.txt'.

Alphanumeric chars: 225

Punctuation chars: 6

Spaces: 56

Other characters: 0

Total characters: 287

Sentences: 3

Words: 59

Letters in words: 225

Factor 1: 381.4

Factor 2: 5.1

Reading level: 5.1

End of Text Turtles

Use file SampleText2 for the final run pasted below.

*[your program code here]\**

**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”s**

**Bold**

*[your program output here]\*\**

//==========================================================

//

// Title: Input File Scanner

// Course: CSC 1101

// Lab Number: Lab 14

// Author: Omar Faruk

// Date: 11/02/2020

// Description:

// Taking an input file and reading the lines, types of chars,

// sentences, words, and factoring while including reading level.

// Used isalnum, ispunct, and loops.

//

//==========================================================

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

int main()

{

// Declare variables

ifstream SampleText\_File;

int line\_count = 0, token\_count = 0, count\_alphanumeric = 0, count\_punctuation = 0, count\_other = 0,

count\_spaces = 0, count\_period = 0, count\_word = 0, letters\_in\_words = 0, count\_total = 0;

string current;

char space = ' ';

double factor\_1=0, factor\_2=0, reading\_level=0;

// Declare constants

const string FILE\_NAME\_INPUT = "SampleText2.txt";

int COLMFT1 = 30;

int COLMFT2 = 7;

// Show application header

cout << "Welcome to Text Turtles!" << endl;

cout << "--------------------------" << endl << endl;

// Attempt to open input file

SampleText\_File.open(FILE\_NAME\_INPUT);

if (!SampleText\_File.is\_open())

cout << "Error: unable to open file '" << FILE\_NAME\_INPUT << "'." << endl << endl;

else

{

while (SampleText\_File.good())

{

int i = 0;

getline(SampleText\_File, current);

//Loop to read from input file

for (i = 0; current[i] != '\0'; i++)

{

// Characters

if (isalnum(current[i]))

count\_alphanumeric++;

else if (ispunct(current[i]))

count\_punctuation++;

else if (current[i] == ' ')

count\_spaces++;

else

{

count\_other++;

}

if (current[i] == '.')

count\_period++;

}

//Line Count

line\_count = line\_count + 1;

}

// Calcutlation

count\_total = count\_alphanumeric + count\_other + count\_punctuation + count\_spaces;

count\_word = count\_spaces + count\_period;

letters\_in\_words = count\_alphanumeric;

factor\_1 = (double)count\_alphanumeric / count\_word \* 100;

factor\_2 = (double)count\_period / count\_word \* 100;

reading\_level = (.0588 \* factor\_1) - (.296 \* factor\_2) - 15.8;

// Format real numbers

cout << fixed << setprecision(1);

// Print Formatted Output

cout << "Reading files from file '" << FILE\_NAME\_INPUT << "' ..." << endl << endl;

cout << line\_count << " line(s) read from file '" << FILE\_NAME\_INPUT << "'." << endl << endl;

cout << setw(COLMFT1) << left << "Alphanumeric chars: "

<< setw(COLMFT2) << right << count\_alphanumeric << endl;

cout << setw(COLMFT1) << left << "Punctuation chars: "

<< setw(COLMFT2) << right << count\_punctuation << endl;

cout << setw(COLMFT1) << left << "Spaces: "

<< setw(COLMFT2) << right << count\_spaces << endl;

cout << setw(COLMFT1) << left << "Other characters: "

<< setw(COLMFT2) << right << count\_other++ << endl;

cout << setw(COLMFT1) << left << "Total characters: "

<< setw(COLMFT2) << right << count\_total << endl << endl;

cout << setw(COLMFT1) << left << "Sentences:"

<< setw(COLMFT2) << right << count\_period << endl;

cout << setw(COLMFT1) << left << "Words:"

<< setw(COLMFT2) << right << count\_word << endl;

cout << setw(COLMFT1) << left << "Letters in words: "

<< setw(COLMFT2) << right << letters\_in\_words << endl << endl;

cout << setw(COLMFT1) << left << "Factor 1:"

<< setw(COLMFT2) << right << factor\_1 << endl;

cout << setw(COLMFT1) << left << "Factor 2:"

<< setw(COLMFT2) << right << factor\_2 << endl;

cout << setw(COLMFT1) << left << "Reading level:"

<< setw(COLMFT2) << right << reading\_level << endl;

// Close input filed

SampleText\_File.close();

}

// Show application close

cout << "\nEnd of Text Turtles!" << endl;

}

