**CSC 1100 – Problem Solving and Programming**

**Project 1 Omar Faruk**

**50 points – Due October 22, 11am**

**Late deadline is October 24, 11:59pm, but 20% off**

**a)** Save this document with your name and the project 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 *Water Wonders* to write a C++ console application that analyzes lake level data. Download text file **MichiganHuronLakeLevels.txt** from Canvas. Place the input file in a folder where your development tool can locate it (on Visual Studio, in folder <project-name>\<project-name>). The input file may be placed in any folder but a path must be specified to locate it. The input file has 158 lines and looks like this:

Lake Michigan and Lake Huron - Average lake levels - 1860-2015

Year Average level (meters)

1860 177.3351667

1861 177.3318333

…

2014 176.3016667

2015 176.59

Within the app, attempt to open the input file and output file **MichiganHuronLakeLevelsHighAndLow.txt**. If the input file didn't open, print an error message. If the output file didn't open, print an error message. Read the input file by scanning past the two header rows. Each detail row in the input file contains two fields (year, lake level). Read one token at a time from the input file. See sample Canvas app **Text file input – one token per read**. Determine the maximum, minimum, and average lake levels. One technique to accomplish this is to use a max variable that starts very small and a min variable that starts very large. After all lines of the input file have been read, use formatted output manipulators (setw, left/right) to print the following rows:

● Column headers.

● Max values.

● Min values.

● Average value.

And columns:

● A left-justified label.

● A right-justified year.

● A right-justified level.

Then write the same information to output file **MichiganHuronLakeLevelsHighAndLow.txt**. Ensure that your code is commented! Provide a complete header comment and body comments. Define constants for the input and output file names and column widths. Format any real numbers to four decimal places. The output should look like this:

Welcome to Wonder Waters

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

Reading lines from file 'MichiganHuronLakeLevels.txt' ...

Writing lines to file 'MichiganHuronLakeLevelsHighAndLow.txt' ...

Year Level (meters)

Max level: … …

Min level: … …

Average level: …

158 line(s) read from file 'MichiganHuronLakeLevels.txt'.

4 line(s) written to file 'MichiganHuronLakeLevelsHighAndLow.txt'.

End of Wonder Waters

*[your program code here]\**

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

//

// Title: Michigan Huron Lake Levels

// Course: CSC 1101

// Author: Omar Faruk

// Date: 10/20/20

// Description:

// Creating a program to display average, max, and min levels

// of water from Lake Huron. Using input to get data from textfile

// and writing the results to an output file.

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

#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 HuronLevels\_File;

ofstream HuronLevels\_FileOutput;

string line;

int line\_count\_in, line\_count\_out;

float average\_level = 0;

int min\_year = 0;

int max\_year = 0;

double level = 0;

double year = 0;

float min\_level = 200.0000;

float max\_level = 0.0000;

// Declare constants

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

const string FILE\_NAME\_OUTPUT = "MichiganHuronLakeLevelsHighAndLow.txt";

const int COLMFT1 = 20;

const int COLMFT2 = 30;

// Format real numbers

cout << fixed << setprecision(4);

// Show application header

cout << "Welcome to Water Wonders!" << endl;

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

// Attempt to open file input

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

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

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

endl;

else

{

// Attempt to open output file

HuronLevels\_FileOutput.open(FILE\_NAME\_OUTPUT);

if (!HuronLevels\_FileOutput.is\_open())

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

else

{

// Print read message

cout << "Reading lines from file '"

<< FILE\_NAME\_INPUT << "'. ..."

<< "\nWriting lines to file '" << FILE\_NAME\_OUTPUT << "'. ..." <<

endl;

line\_count\_in = 2;

line\_count\_out = 0;

while (HuronLevels\_File.good())

{

//Reading past header comments

getline(HuronLevels\_File, line);

getline(HuronLevels\_File, line);

//Read x,y

while (HuronLevels\_File >> year >> level)

{

//Min Output

if (level < min\_level)

{

min\_year = year;

min\_level = level;

}

//Max Output

if (level > max\_level)

{

max\_year = year;

max\_level = level;

}

//Average Cost

average\_level += level;

//Line counter

line\_count\_in = line\_count\_in + 1;

}

average\_level = average\_level / (line\_count\_in -2);

}

}

// Print to screen

cout << setw(COLMFT2) << left << " "

<< setw(COLMFT1) << right << "Year"

<< setw(COLMFT1) << right << "Level (meters)" << endl;

cout << setw(COLMFT2) << left << "Max Level:"

<< setw(COLMFT1) << right << max\_year

<< setw(COLMFT1) << right << max\_level << endl;

cout << setw(COLMFT2) << left << "Min Level:"

<< setw(COLMFT1) << right << min\_year

<< setw(COLMFT1) << right << min\_level << endl;

cout << setw(COLMFT2) << left << "Average Level:"

<< setw(COLMFT1) << right << ""

<< setw(COLMFT1) << right << average\_level << endl;

// Write to output file

HuronLevels\_FileOutput << setw(COLMFT2) << left << " "

<< setw(COLMFT1) << right << "Year"

<< setw(COLMFT1) << right << "Level (meters)" << endl;

HuronLevels\_FileOutput << setw(COLMFT2) << left << "Max Level:"

<< setw(COLMFT1) << right << max\_year

<< setw(COLMFT1) << right << max\_level << endl;

HuronLevels\_FileOutput << setw(COLMFT2) << left << "Min Level:"

<< setw(COLMFT1) << right << min\_year

<< setw(COLMFT1) << right << min\_level << endl;

HuronLevels\_FileOutput << setw(COLMFT2) << left << "Average Level:"

<< setw(COLMFT1) << right << ""

<< setw(COLMFT1) << right << average\_level << endl;

line\_count\_out = 4;

// Close input file

HuronLevels\_File.close();

cout << endl << line\_count\_in

<< " line(s) read from file '"

<< FILE\_NAME\_INPUT << "'." << endl;

// Close output file

HuronLevels\_FileOutput.close();

cout << line\_count\_out << " line(s) written to file '"

<< FILE\_NAME\_OUTPUT << "'." << endl << endl;

}

// Show application close

cout << "\nEnd of Water Wonders" << endl;

}

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



