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

**Lab 12 – rory lange**

**25 points – Due March 8, 11pm**

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

**b)** Type/paste your answers into the document.

**c)** Submit the following documents to the Canvas assignment link where you downloaded this document:

✓ This document.

✓ Your .cpp files renamed to .txt.

Submit the documents separately, not as one .zip file.

**1) [13 points]** You've been hired by *Faraway Fantasies* to write a C++ console application that reads data from file **NearestStars.txt**. The file contains the ten nearest stars to Earth. It has a header line and ten detail lines. Each detail line contains two tokens:

● Star name

● Distance from Earth in light minutes (Sun) and light years (all other stars).

Place the file in a folder where your development tool can locate it, or place the file in a data folder and use a path to the file. In the app, attempt to open the file. If it doesn't open, print an error message. If it does open, print a file-reading message, and three column headers:

● Star

● Distance

● Time (described next)

Use a while loop to read through the file. For each detail line, read the star name and distance. Assume there is a spacecraft that travels 0.4 light years per hour (yes, that's fast; is it possible? Ask Jacob.). Calculate the time to travel to the star. Note that the distance to the Sun is in light minutes so the time calculation will have to use a factor of 525,600 minutes per year. Use formatted output manipulators (setw, left/right) to print in one row the following:

● Star

● Distance

● Time

Track the number of lines read and print the count after the loop. Define constants for the input file name, spacecraft speed, minutes per year, and the column widths. Format all real numbers to six decimal places. See sample app **TextFileInput-OneTokenPerRead.cpp** on Canvas. The output should look like this:

Welcome to Faraway Fantasies

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

Reading tokens from file 'NearestStars.txt' ...

Star Distance\* Time (hours)

…

\*Distance to sun is in light minutes. Distance to all other stars is in light years.

11 line(s) read from file 'NearestStars.txt'.

End of Faraway Fantasies

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

//

// Title: Faraway Fantasies

// Course: CSC 1101

// Lab Number: 12-1

// Author: Rory Lange

// Date: 3/4/21

// Description:

// This C++ console application reads data from file

// NearestStars.txt. The file contains the ten nearest

// stars to Earth. It has a header line and ten detail

// lines. Each detail line contains two tokens:

// -Star name

// -Distance from Earth in light minutes(Sun) and light

// years(all other stars).

// It attempts to open the file. If it doesn't open, it

// prints an error message. If it does open, it prints a

// file-reading message, and three column headers:

// -Star

// -Distance

// -Time

// The application reads through the file. For each

// detail line, it reads the star name and distance. It

// calculates the time to travel to the star. It then

// prints:

// -Star

// -Distance

// -Time

//

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

#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()

{

const string fileName = "NearestStars.txt";

const int w = 20;

const int w1 = 15;

int lines = 1;

string star;

double distance;

double time;

double speed = .4;

double sunSpeed = 210240;

//header

cout << "Faraway Fantasies File Reader" << endl;

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

cout << fixed << setprecision(6);

int line;

string text;

ifstream file;

file.open(fileName);

if (!file.is\_open())

cout << "Error: File did not open properly" << endl;

else {

getline(file, text);

while (file.good()) {

file >> star;

file >> distance;

time = (star == "Sun") ? distance / sunSpeed : distance / speed;

cout << left << setw(w) << star << right << setw(w1) << distance << right << setw(w1) << time << " (hours)" << endl;

lines++;

}

cout << endl << "\*distance to the sun is in light minutes. All other starts in light years" << endl << endl;

cout << lines << " lines read from file " << fileName << endl << endl;

}

cout << "End of calculations for Faraway Fantasies.";

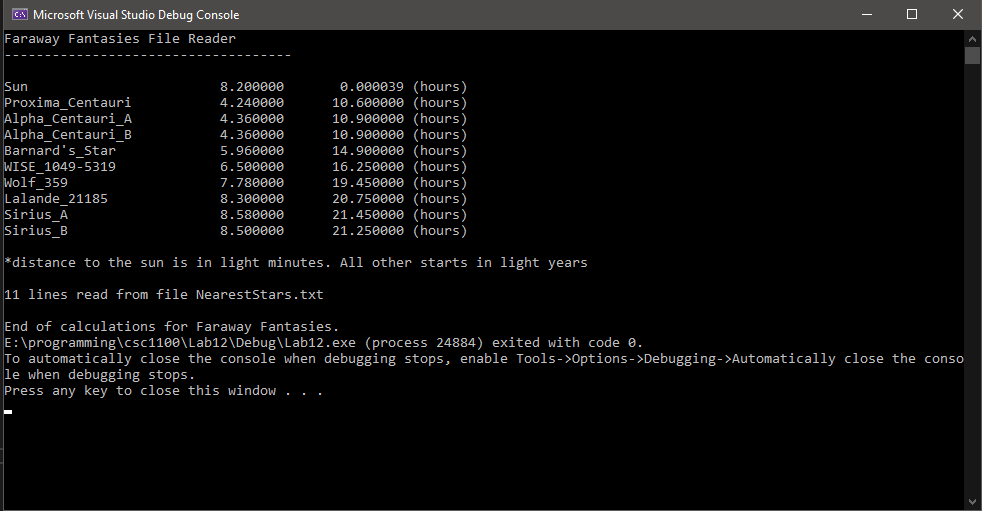
}

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

**Font “Courier New”**

**Font size “9”**

**Bold**



**2) [12 points]** You've been hired by *Utterly Urban* to write a C++ console application that prompts the user for city names and writes them to file **Cities.txt**. The file will contain one city per line. It does not have a header line, only detail lines. The file will be written to the default folder of your development tool unless you specify a path to the file. In the app, attempt to create (open) the file. If it doesn't open, print an error message. If it does open, print a file-reading message. Use a sentinel loop to continue prompting the user for city names until they enter the sentinel value (of your choosing). If the sentinel value wasn't entered, prompt for and get from the user a city name and write it to one line of the file. Track the number of cities written and print the count after the loop. Define a constant for the output file name. See sample app **TextFileOutput.cpp** on Canvas. The output should look like this:

Welcome to Utterly Urban

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

Writing lines to file 'Cities.txt' ...

Enter another city (y/n)? y

Enter a city (no spaces): Trenton

Enter another city (y/n)? y

Enter a city (no spaces): Riverview

Enter another city (y/n)? n

2 cities written to file 'Cities.txt'.

End of Utterly Urban

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

//

// Title: Utterly Urban

// Course: CSC 1101

// Lab Number: 12-2

// Author: rory lange

// Date: 3/4/21

// Description:

// This C++ console application that prompts the user for

// city names and writes them to file Cities.txt. The file

// will contain one city per line. It does not have a

// header line, only detail lines. It attempts to create

// (open) the file. If it doesn't open, the application

// prints an error message. If it does open, it prints a

// file-reading message. It uses a sentinel loop to

// continue prompting the user for city names until they

// enter the sentinel value. If the sentinel value wasn't

// entered, it prompts for and gets from the user a city

// name and writes it to one line of the 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()

{

//header

cout << "Welcome to Utterly Urban" << endl;

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

//declare variables

string input;

const string fileName = "cities.txt";

ofstream file;

char go = 'y';

int counter = 0;

cout << "writing lines to file " << fileName << " ..." << endl << endl;

file.open(fileName);

//sentinel

while (go != 'n') {

//get inputs

cout << "Enter a city (no spaces): ";

cin >> input;

if (!file.is\_open()) {

cout << "Error opening file." << endl;

break;

}

else {

file << input << "\n";

}

counter++;

//prompt for sentinel

cout << "Would you like to enter another city (y/n)?";

cin >> go;

}

cout << counter << " cities were entered." << endl;

//end

cout << "End of calculations" << endl;

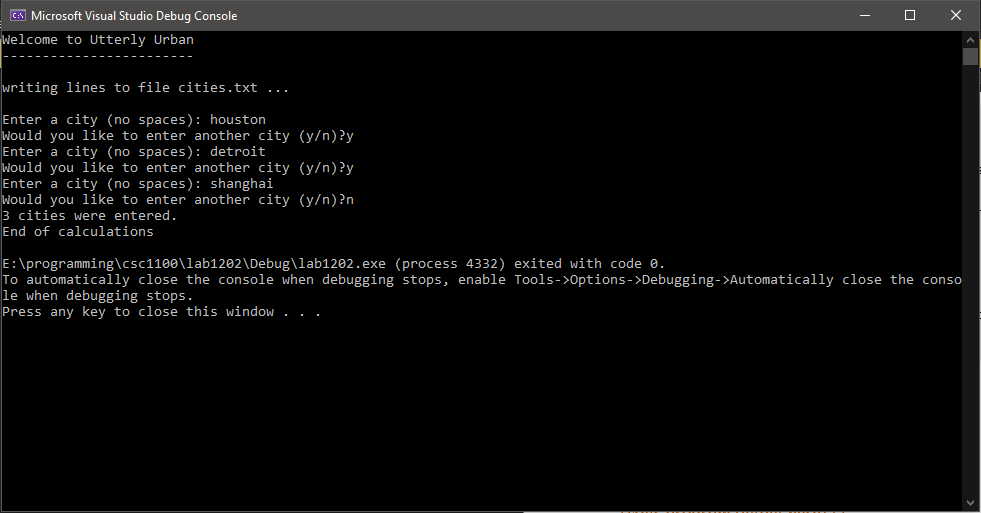
}

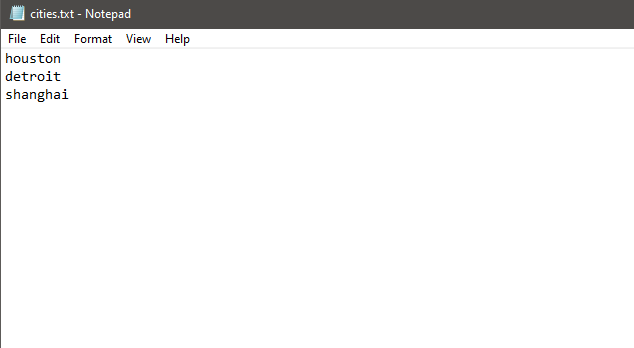
**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”**

**Bold**

**



**\* Copying-and-pasting C++ code to a Word document**

**macOS**

1) From within the C++ program, press **command-A** and press **command-C**.

2) From within the Word document, press **command-V**.

**Windows**

1) From within the C++ program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

**\*\* Copying-and-pasting C++ console application output to a Word document**

**macOS**

1) From the C++ console, press **shift-command-4-space**.

2) From within the Word document, **command-V**.

**Windows**

1) From the C++ console, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.