# Assignment Description

For this assignment, you will analyze the results of some ghost hunts.  You will read data from ghosts.txt (a sample is provided in the starter code). The file contains evidence collected during a recent ghost hunt. There will be three events in a file, 1 per line. Each line will have the event code and the number of times that event occurred.  As you read in the event, you will need to calculate an overall haunting score for the ghost hunt.  If the haunting score is greater than 15, then you should display the message "This place is haunted." Otherwise, the program should display "This place is probably not haunted." Before the program finishes, you should print the message "Great job! On to the next one."

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/d3c3b216b5b106f942cf96f9f5c9664151396fa4/M03%20Programming%20Assignment%202](https://github.com/wesleyhixon/Programming-Assignments/tree/d3c3b216b5b106f942cf96f9f5c9664151396fa4/M03 Programming Assignment 2)

# Readme Documentation

Input Information: Input is a file containing a series of ghost event tags followed by the number of times that event occurred

Output Information: Output is the number of instances of each event found, followed by the place’s haunting score, followed by whether the place is haunted.

# Flowchart Screen Shots (optional)

Screen shot(s) here

# UML and Use Case Diagrams (optional)

Screen shot(s) here

# Source Code of All files (.h, .cpp)

/\*

Program Name: Ghost Hunting

Author: Wesley Hixon

Date Last Updated: 06/19/2024

Purpose: Analyze an input file for ghost events and determine whether a place is haunted

\*/

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

using namespace std;

int main(){

// Declaring variables

string filename = "ghosts.txt";

// (I had to set hauntingScore to 0 because without it hauntingScore initializes at around 30,000)

int shadowsOrOrbs, EVP, footsteps, fullBodyApparition, objectsMoving, doorsShutting, thermoCamera, hauntingScore = 0;

// Outputting status

cout << "Analyzing Ghost Hunt Data ... \n";

// Opening ghosts.txt

ifstream ghosts;

ghosts.open(filename);

string line;

// Looping until there are no more lines

while(getline(ghosts, line)){

// Declaring stringstream variable s in order to manipulate the line as an input stream

// Also declaring eventCode to use in switch

char eventCode;

stringstream s(line);

s >> eventCode;

// Each switch case is a character symbolizing each ghost hunting event

switch(eventCode){

case 'E':

// Extract the number of events from the stringstream variable

s >> EVP;

// Add the event number times its score to the overall haunting score

hauntingScore += (EVP \* 3);

// Output how many events were found

cout << "Found " << EVP << " instances of EVP (Electronic Voice Phenomena). \n";

break;

case 'S':

s >> shadowsOrOrbs;

hauntingScore += shadowsOrOrbs;

cout << "Found " << shadowsOrOrbs << " instances of shadows or orbs. \n";

break;

case 'F':

s >> footsteps;

hauntingScore += (footsteps \* 2);

cout << "Found " << footsteps << " instances of footsteps. \n";

break;

case 'A':

s >> fullBodyApparition;

hauntingScore += (fullBodyApparition \* 5);

cout << "Found " << fullBodyApparition << " instances of a full body apparition. \n";

break;

case 'O':

s >> objectsMoving;

hauntingScore += (objectsMoving \* 5);

cout << "Found " << objectsMoving << " instances of objects moving on their own. \n";

break;

case 'D':

s >> doorsShutting;

hauntingScore += (doorsShutting \* 3);

cout << "Found " << doorsShutting << " instances of doors shutting on their own. \n";

break;

case 'T':

s >> thermoCamera;

hauntingScore += (thermoCamera \* 2);

cout << "Found " << thermoCamera << " instances of ThermoCamera hot or cold spots.\n";

break;

}

}

// Output haunting score

cout << "The haunting score for this place is " << hauntingScore << ". \n";

// Depending on the haunting score, output whether the place is haunted

if(hauntingScore > 15){

cout << "This place is haunted. \n";

}

else{

cout << "This place is probably not haunted.";

}

ghosts.close();

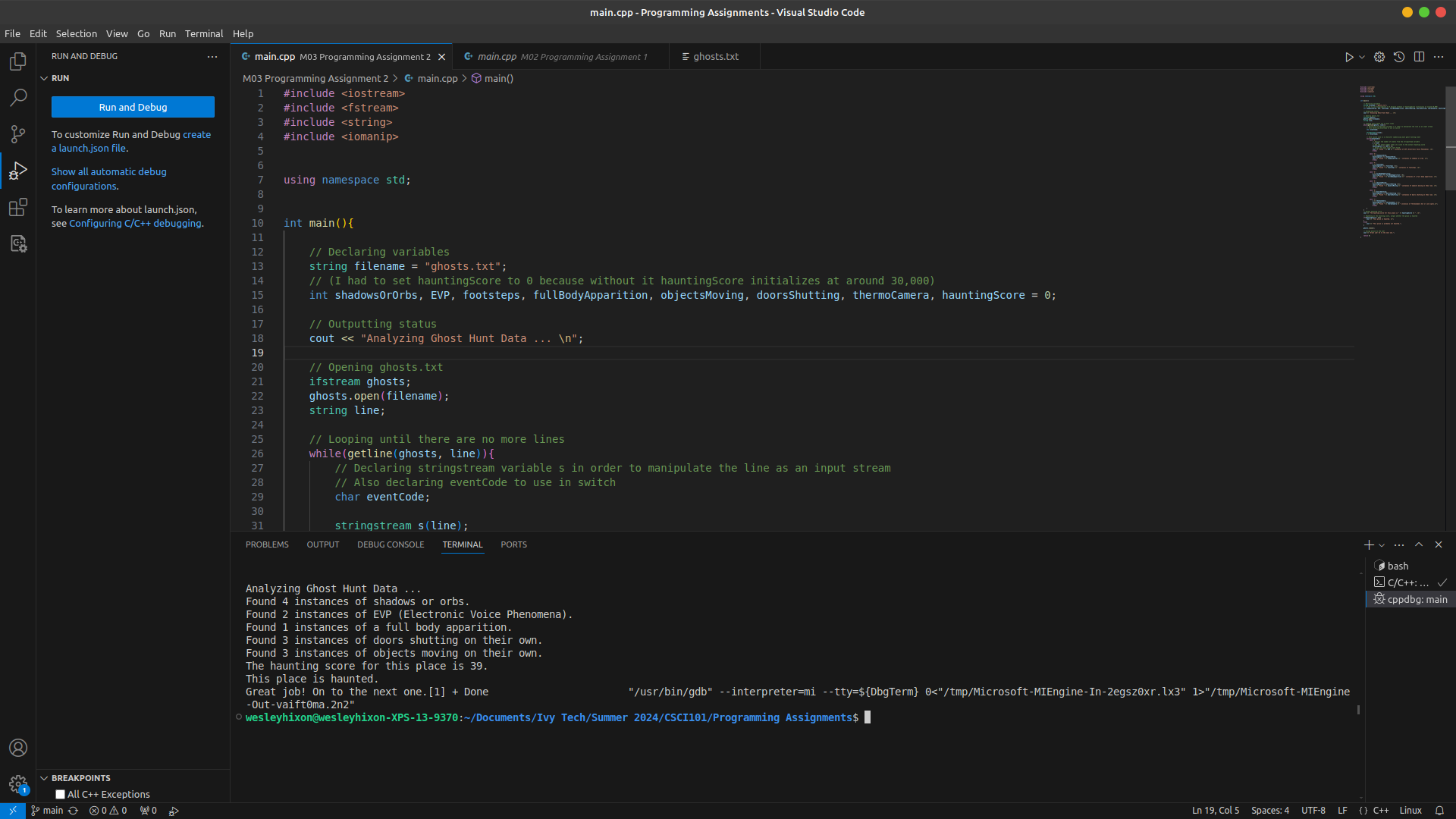
// Assign praise to the user

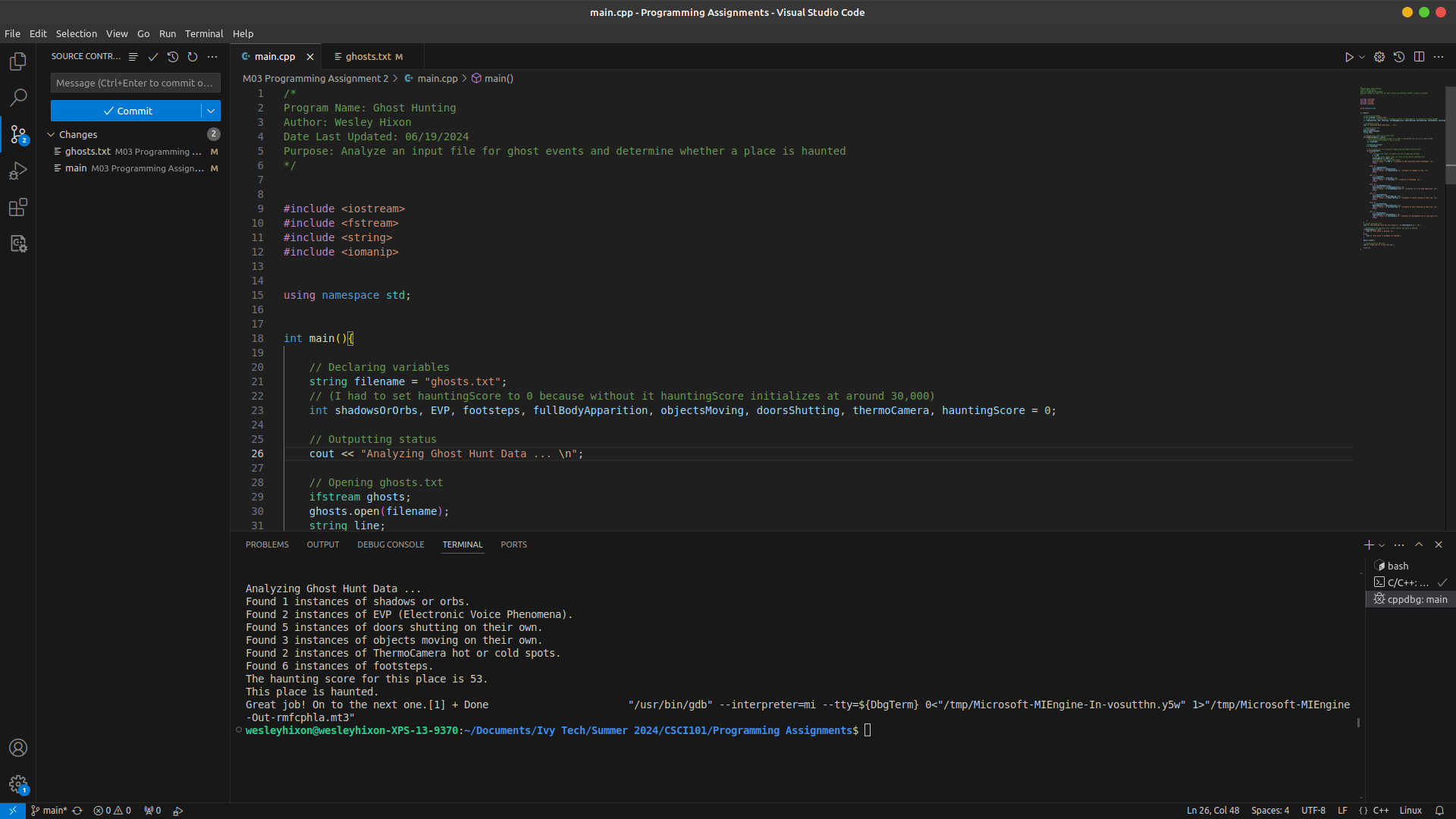
cout << "Great job! On to the next one.";

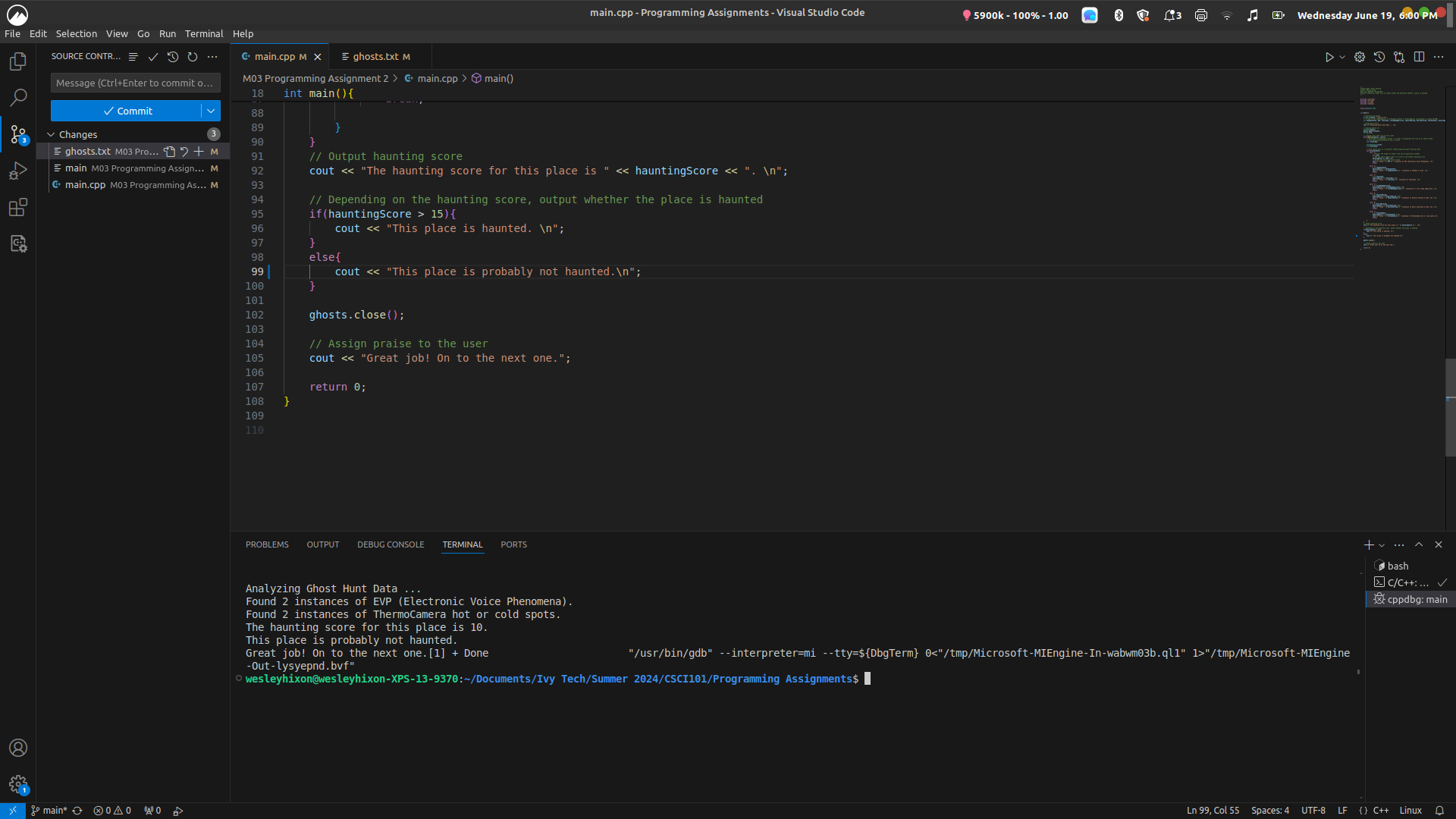
return 0;

}

# Three Use Case Screen Shots







Please make sure all screen shots and text are clearly viewable for faster / easier grading!