# Assignment Description

Write a program in C++ that asks the user 5 binary digits. The digits will represent a binary number.  Convert the binary number into decimal using the following formula: digit5 + digit4\*2 + digit3\*4 + digit2\*8 + digit1\*16

# GitHub URL (optional)

# Readme Documentation

Input Information: Input is 5 binary digits

Output Information: Output is the binary number converted to a decimal number

# 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: Binary to Decimal Conversion

\* Author: Wesley Hixon

\* Date last updated: 6/6/2024

\* Purpose: To convert a 5 digit binary number to a decimal number

\*/

#include <iostream>

#include <string>

using namespace std;

int main()

{

// Declaring variables

int digit1, digit2, digit3, digit4, digit5;

string binary\_str;

int decimal\_num;

// Prompting the user for input of binary digits

cout << "Enter the 1st binary digit: ";

cin >> digit1;

cout << "Enter the 2nd binary digit: ";

cin >> digit2;

cout << "Enter the 3rd binary digit: ";

cin >> digit3;

cout << "Enter the 4th binary digit: ";

cin >> digit4;

cout << "Enter the 5th binary digit: ";

cin >> digit5;

// Converting the binary digits to a decimal number

decimal\_num = digit5 + (digit4 \* 2) + (digit3 \* 4) + (digit2 \* 8) + (digit1 \* 16);

// Converting the binary digits to strings so that they can be concatenated

string str5 = to\_string(digit5);

string str4 = to\_string(digit4);

string str3 = to\_string(digit3);

string str2 = to\_string(digit2);

string str1 = to\_string(digit1);

// Creating the binary string

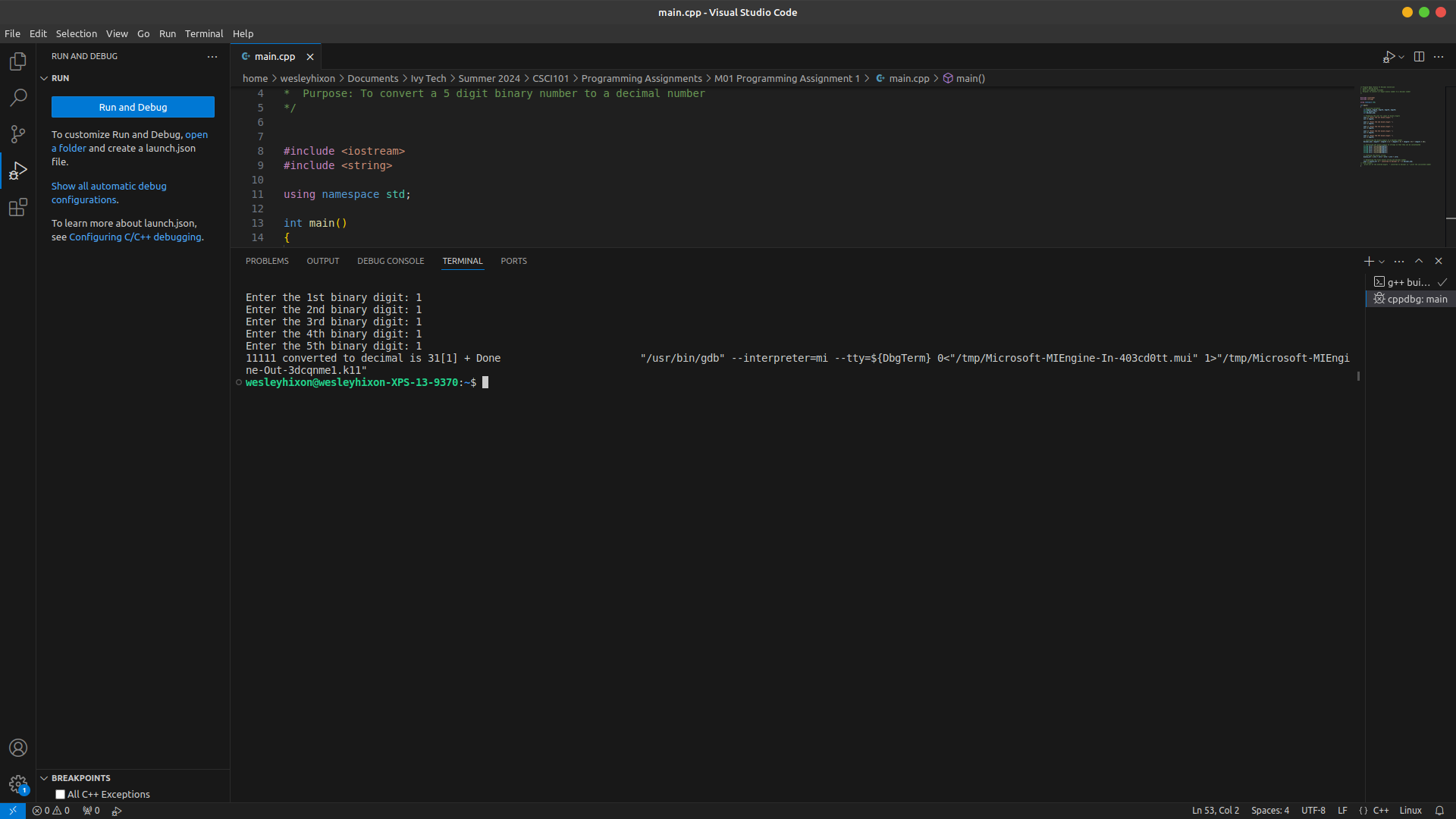
binary\_str = str1 + str2 + str3 + str4 + str5;

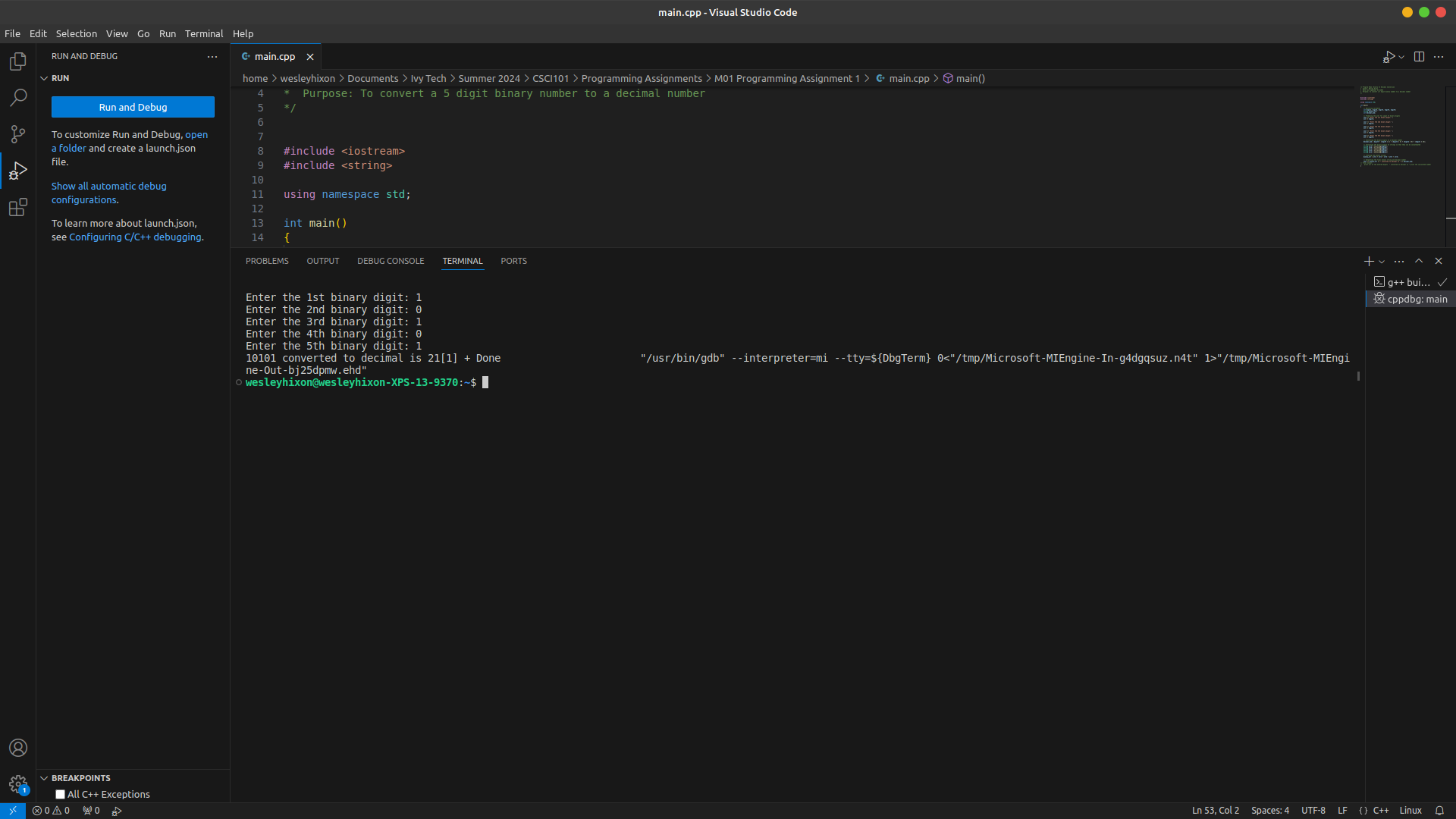
// Outputting the final binary string and decimal number

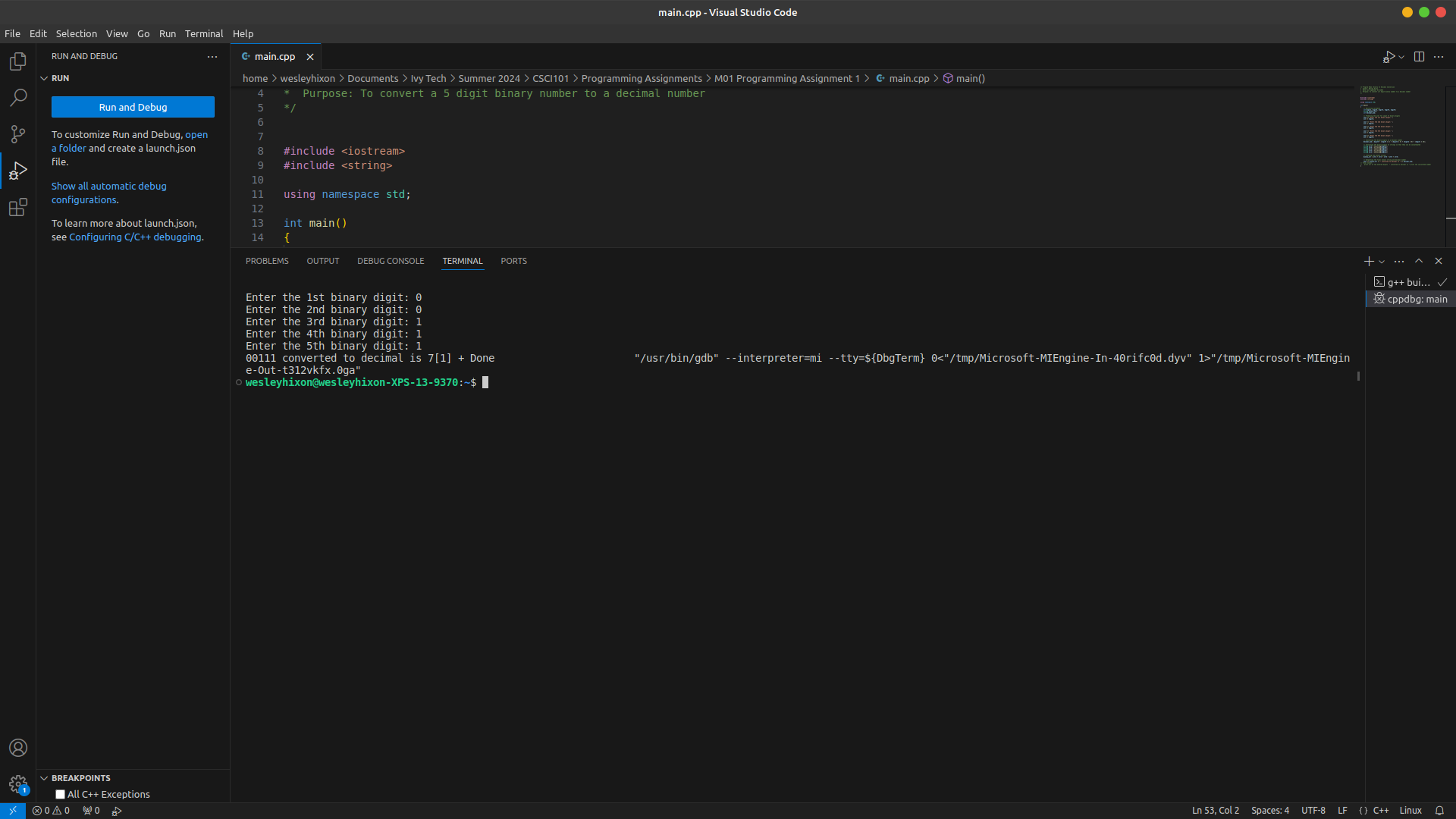
cout << binary\_str << " converted to decimal is " << decimal\_num;

}

# Three Use Case Screen Shots

The can all be in a single screen shot





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