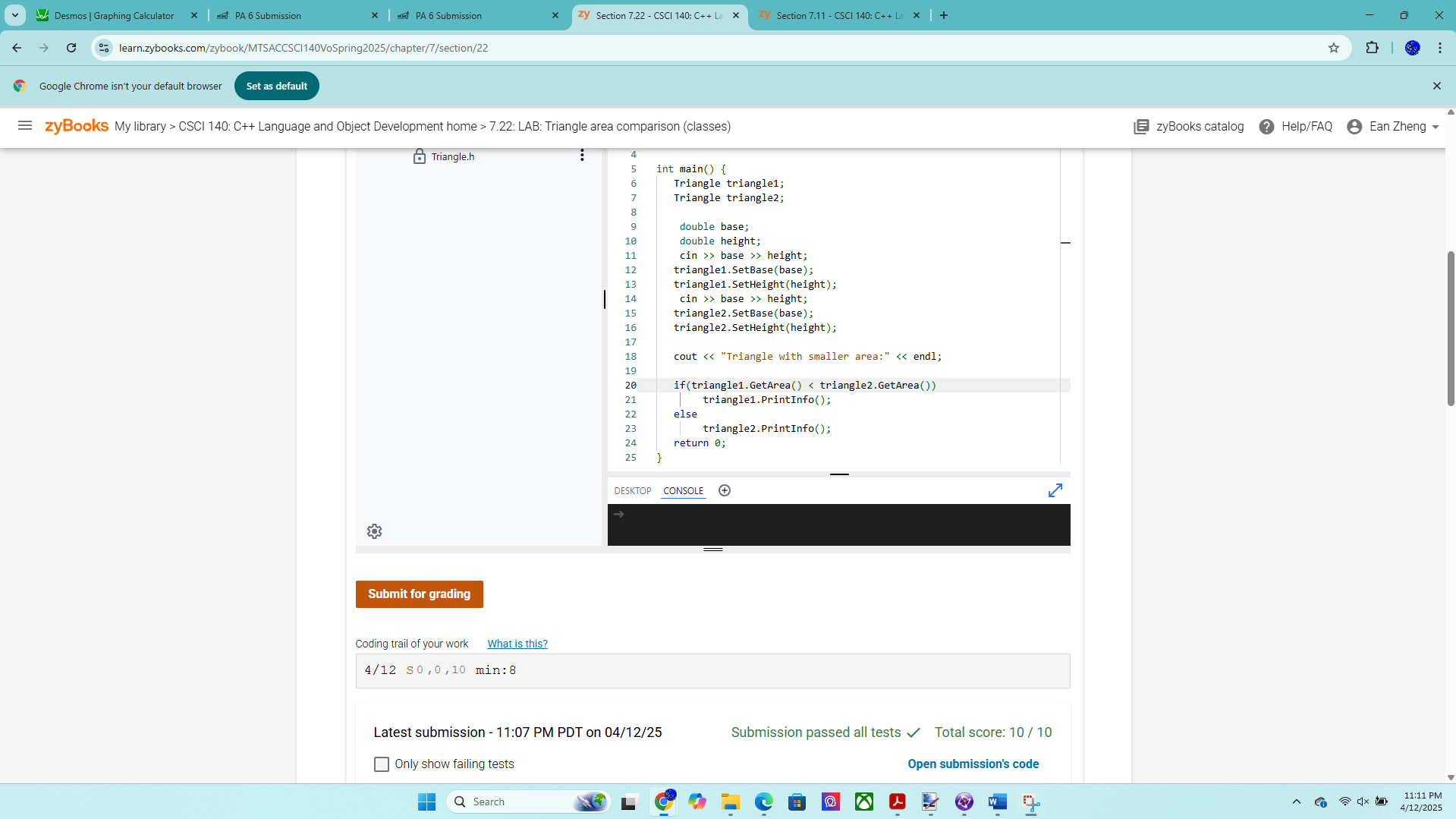
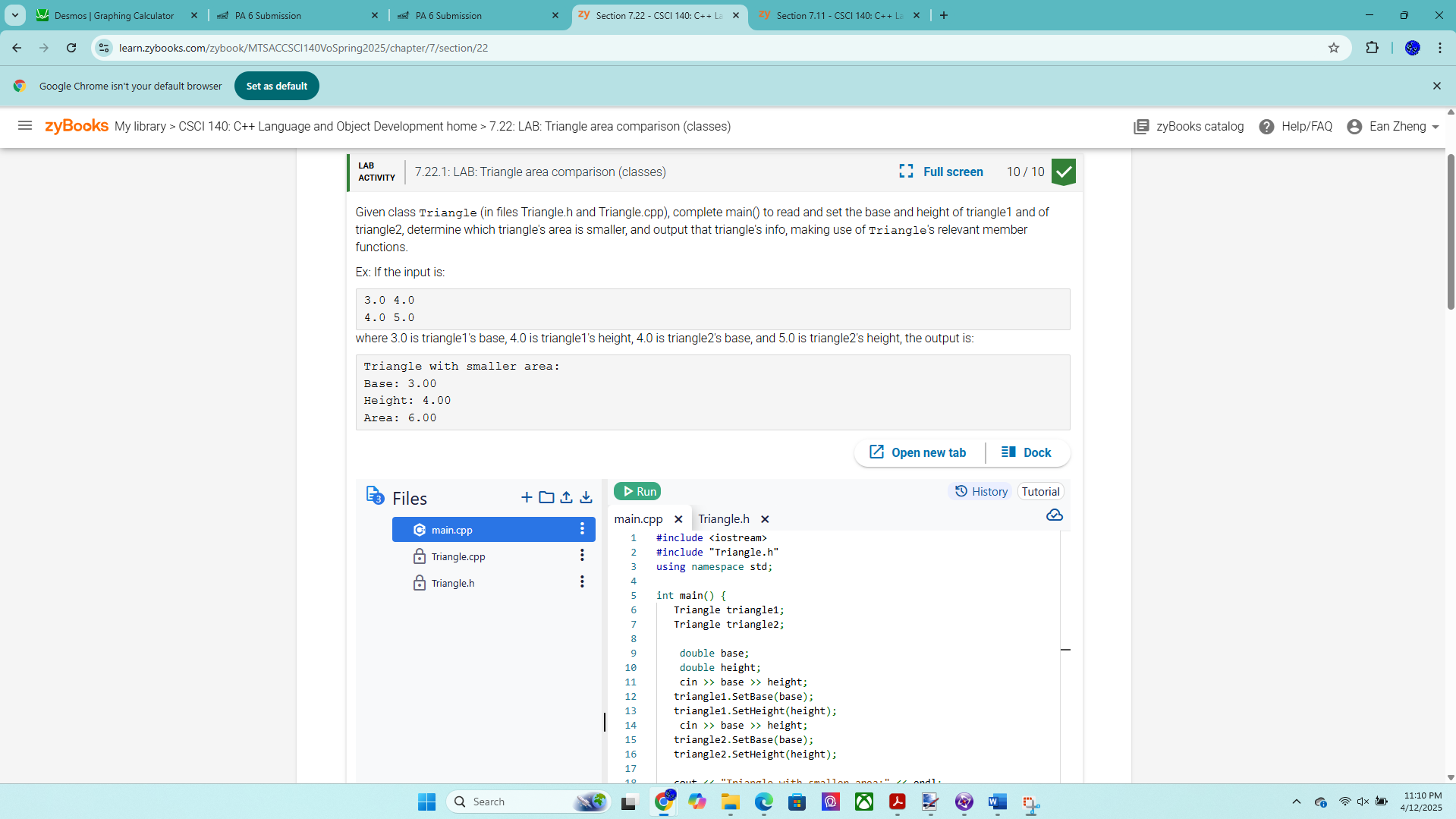
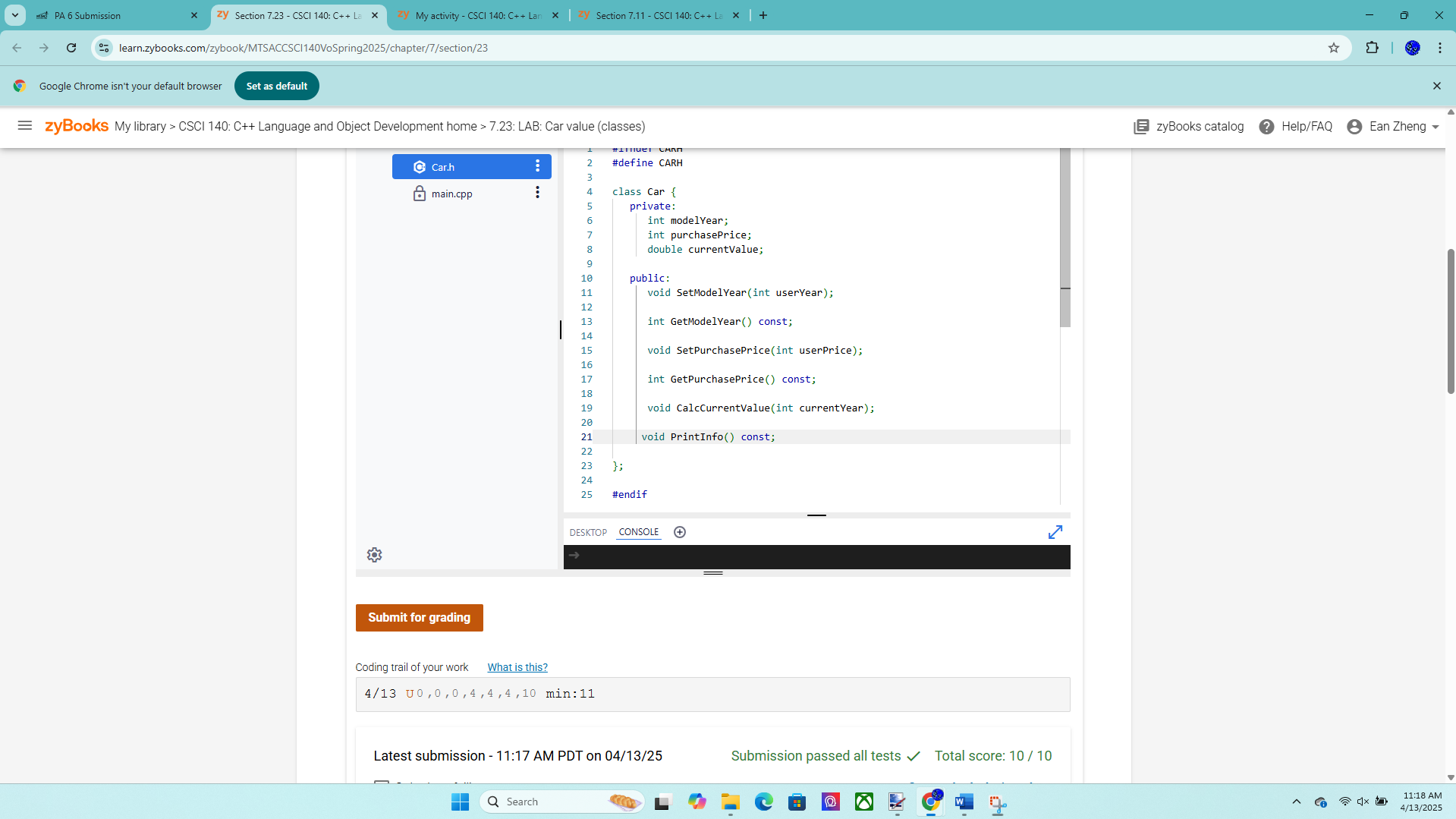
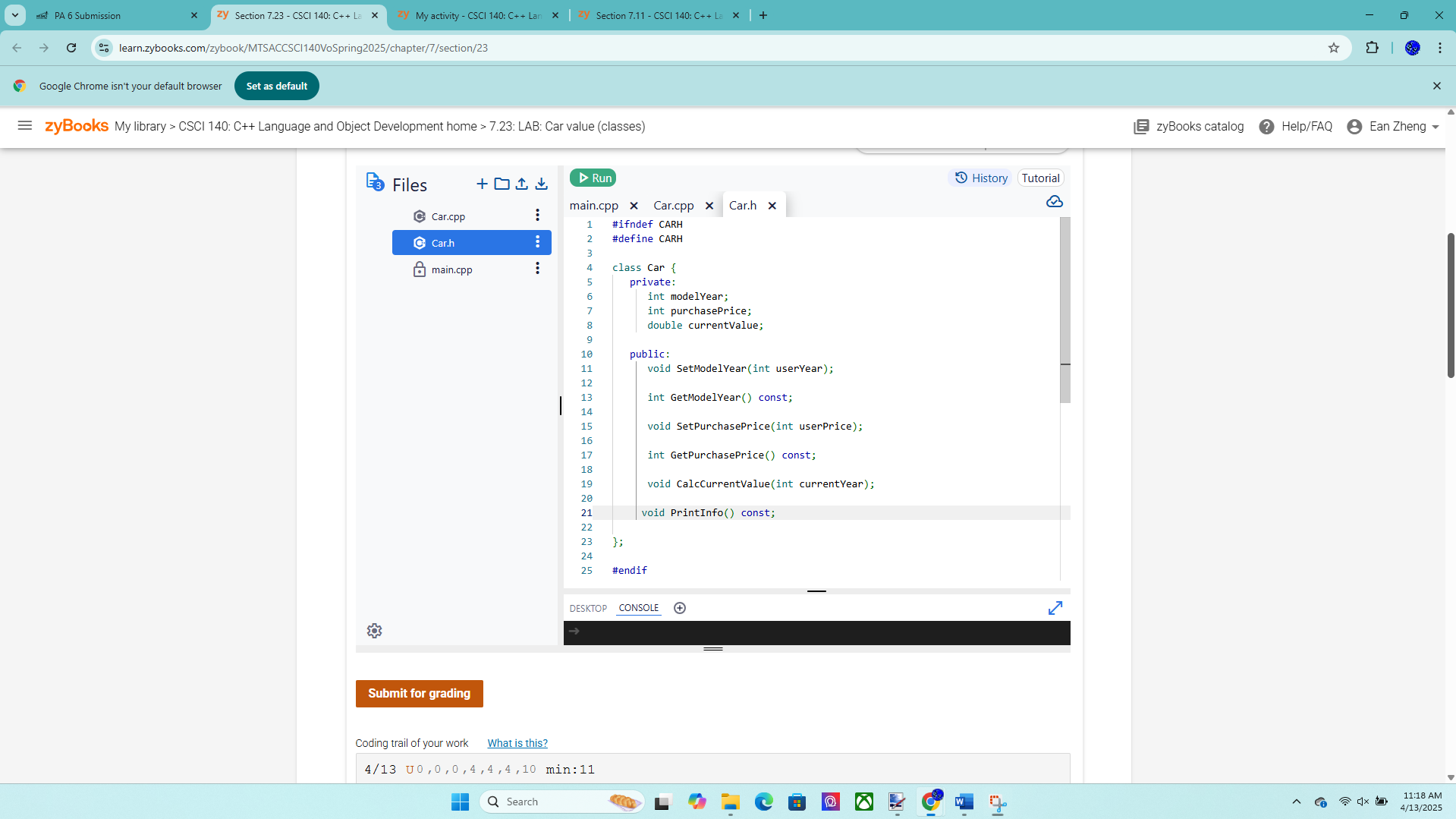
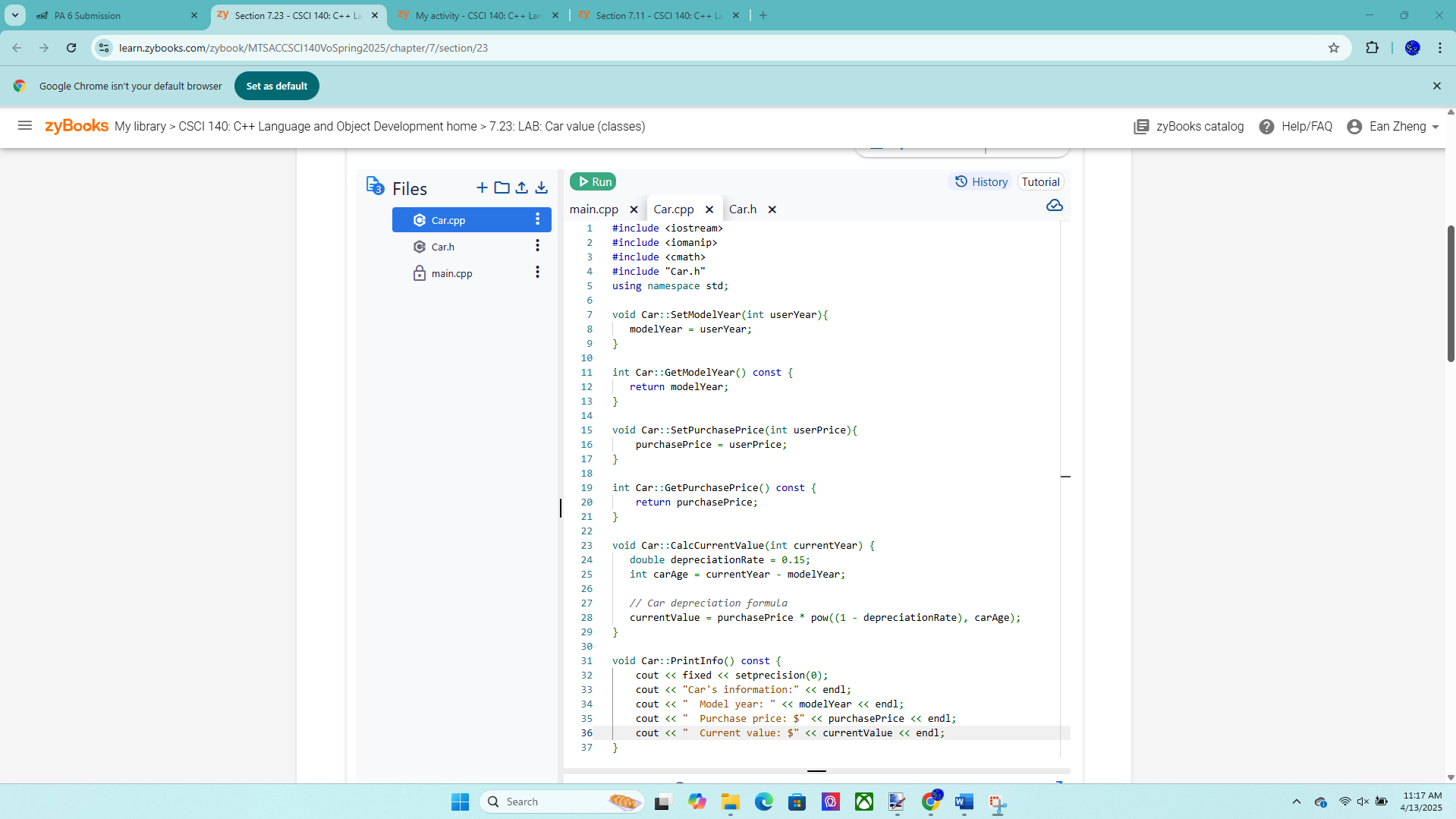
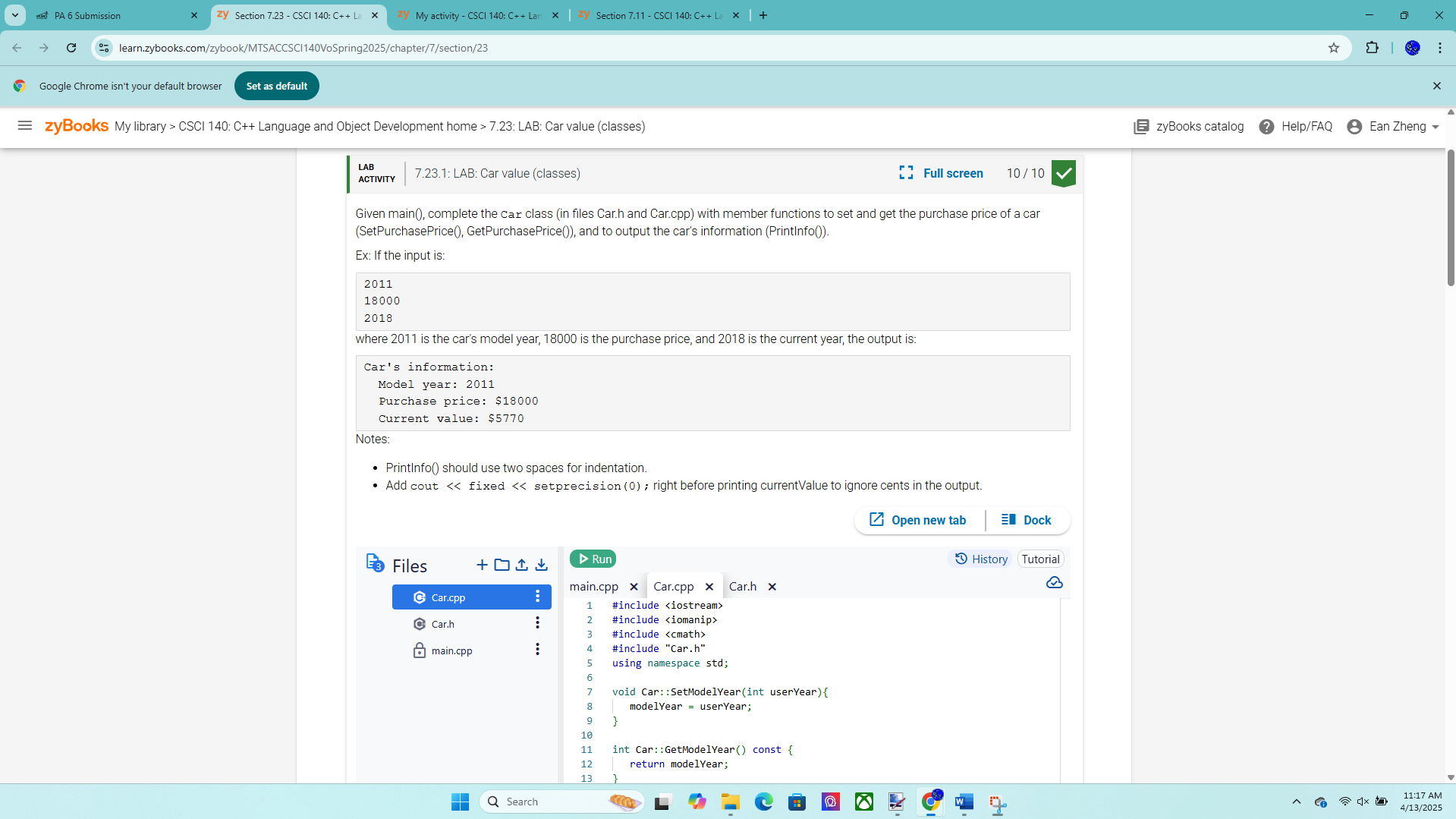
## CSCI 140 PA 6 Submission

## Due Date: 4/14/2025 Late (date and time):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

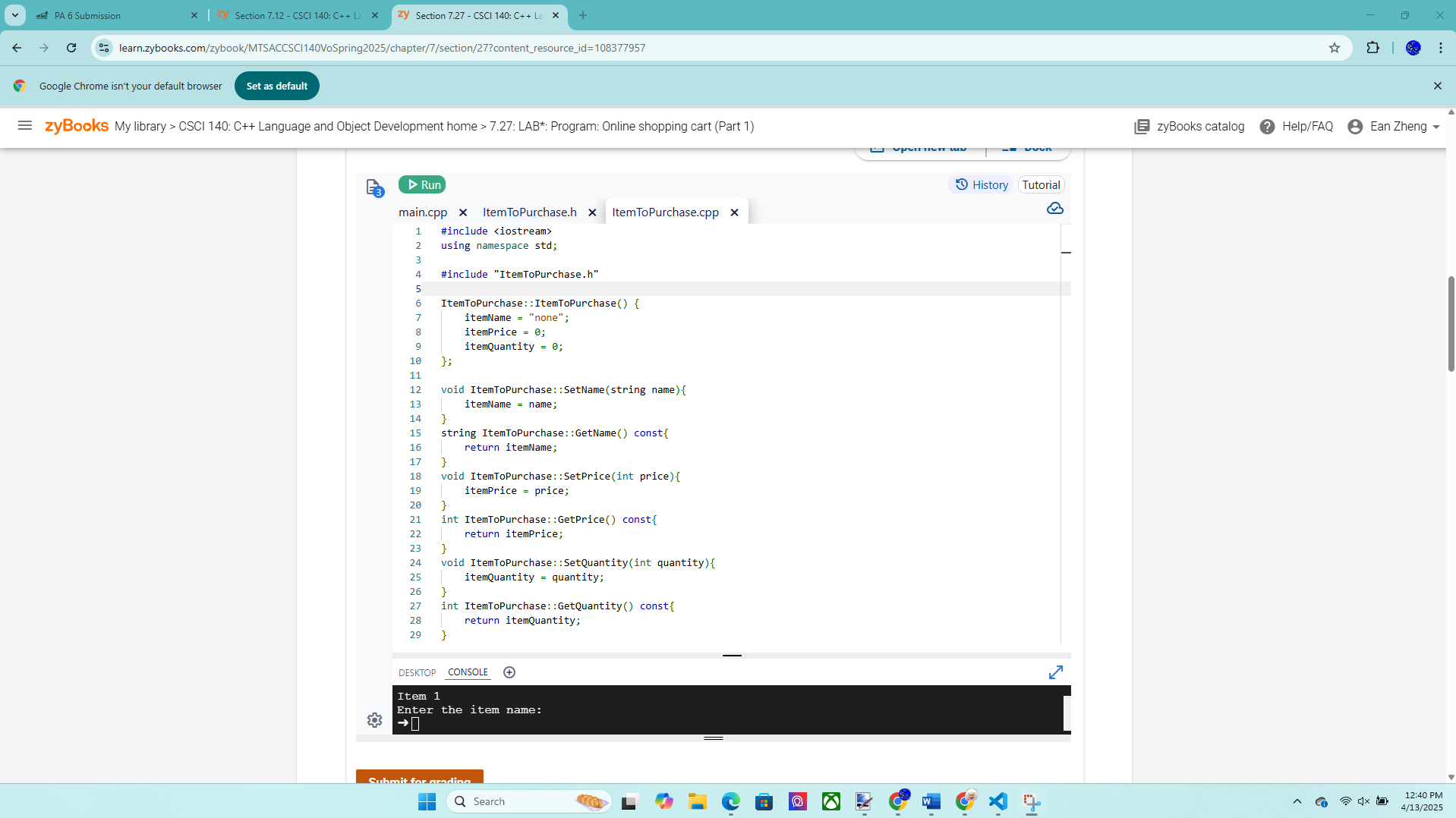
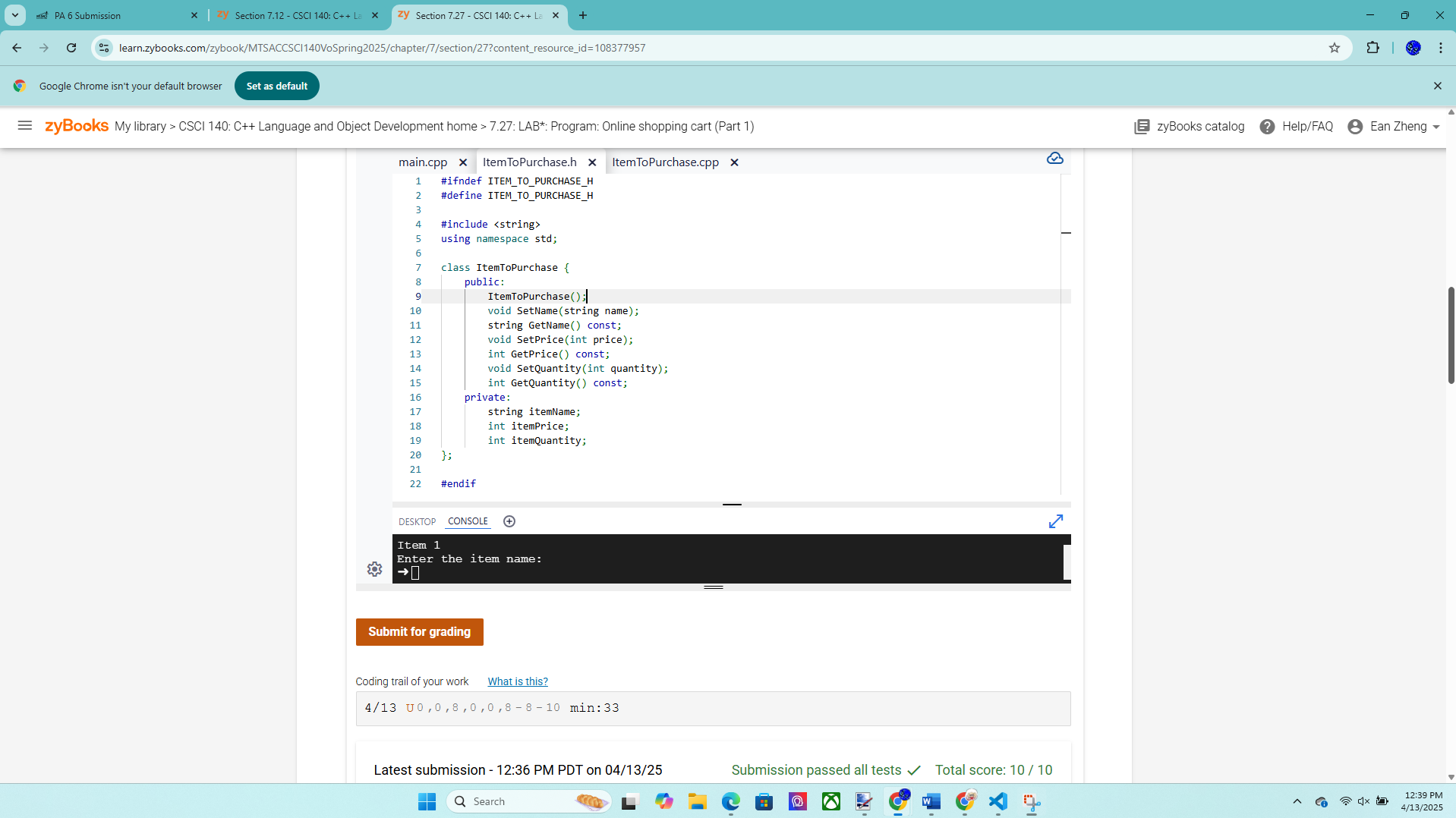
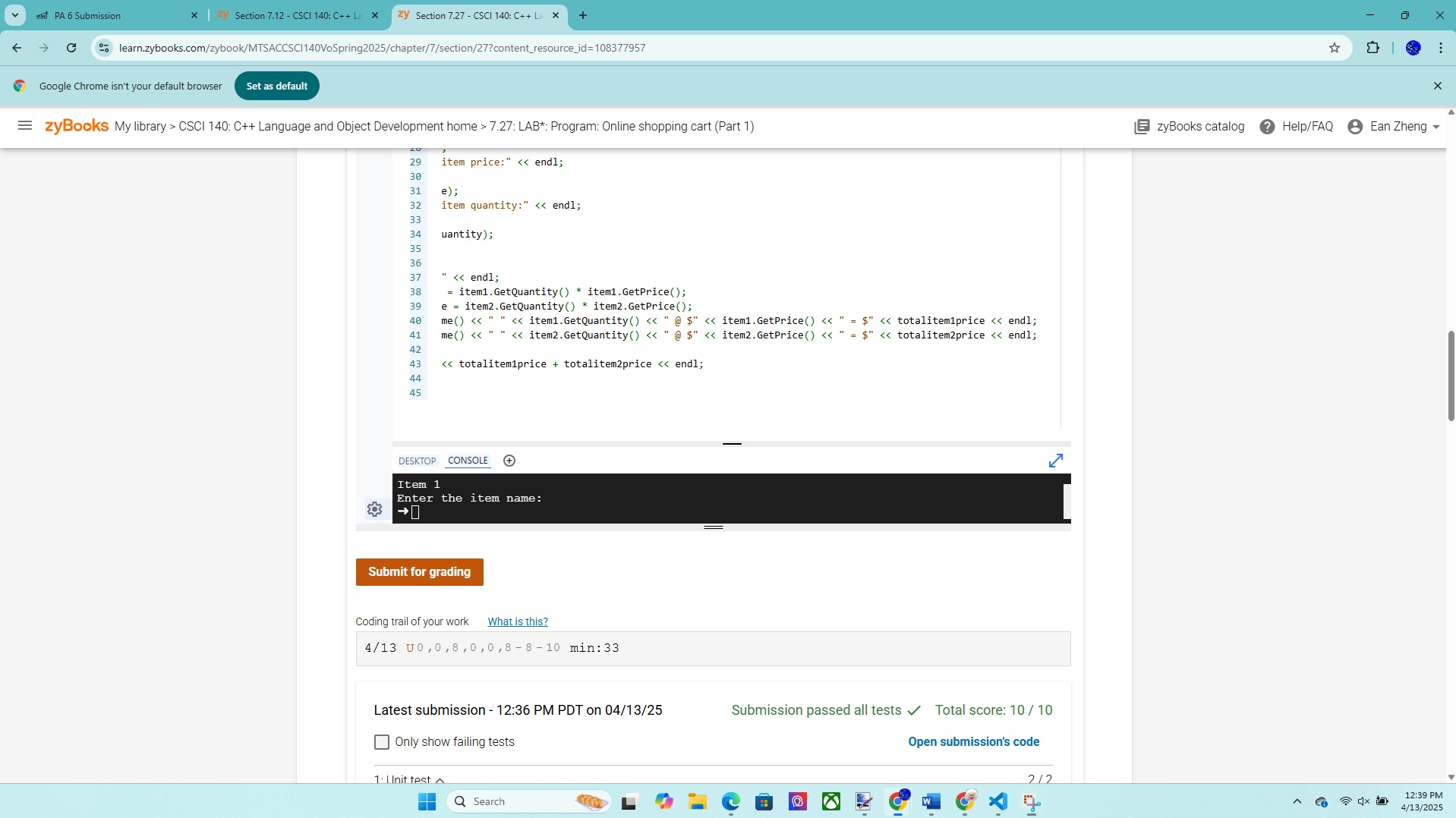
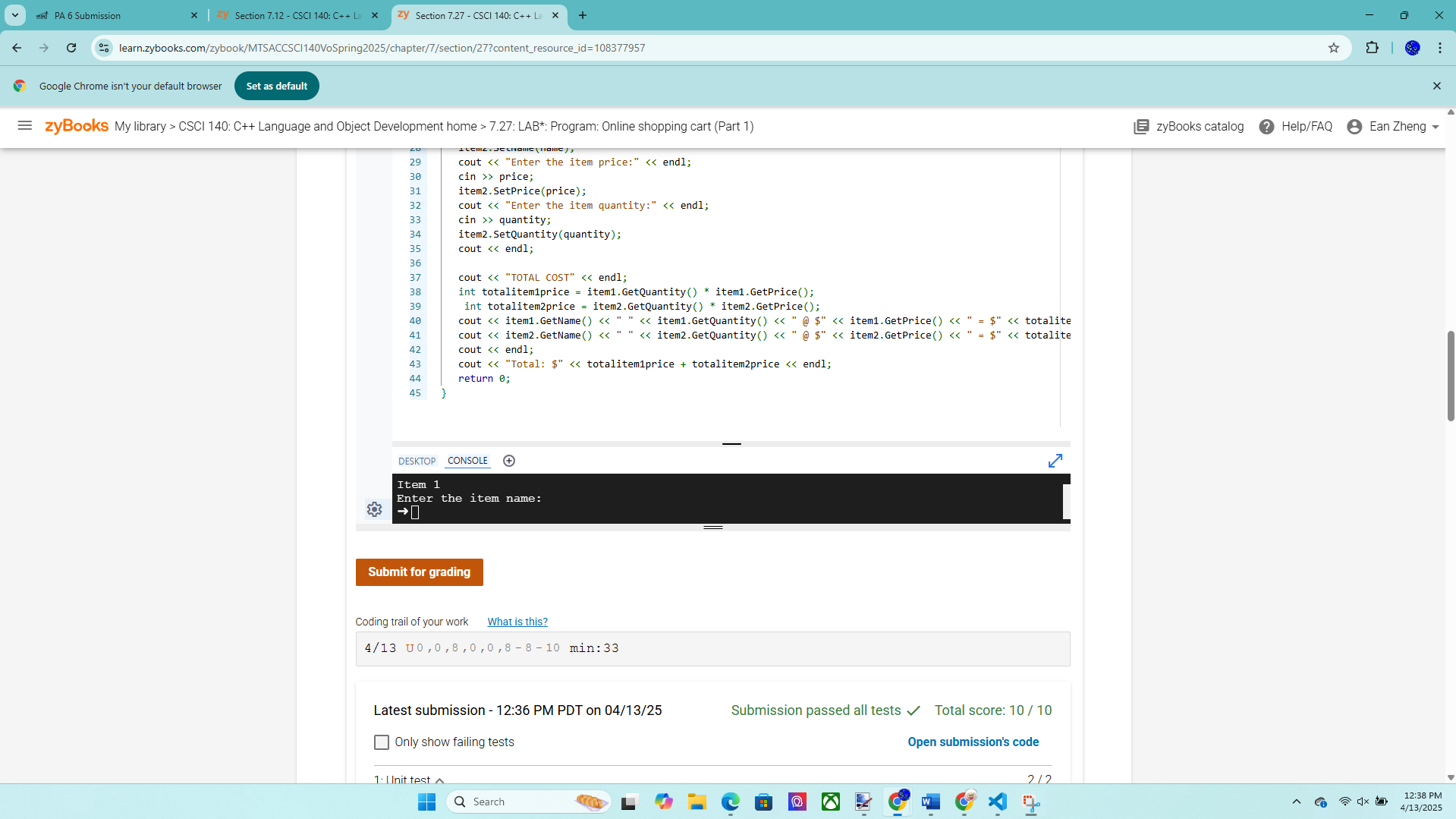
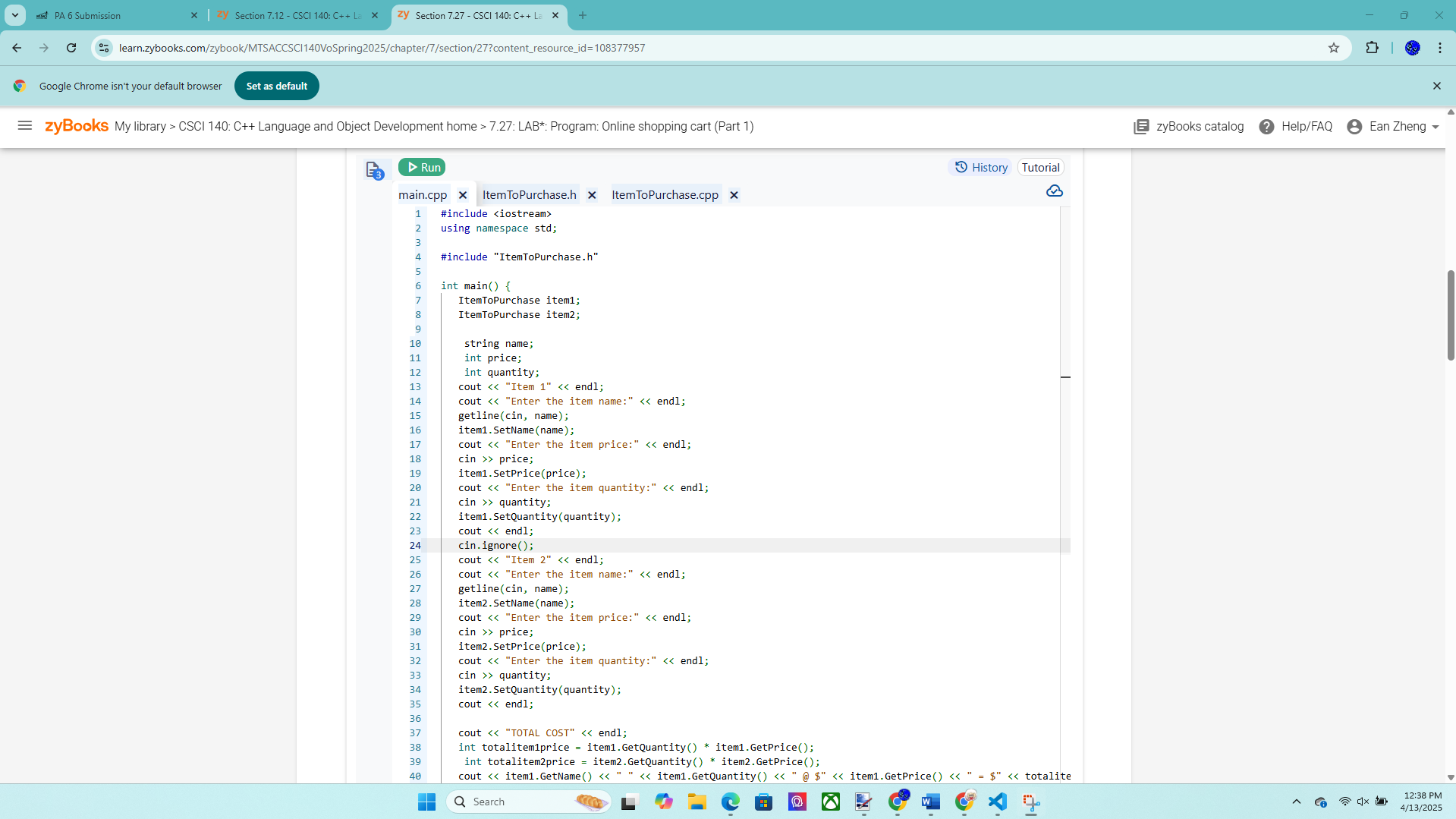
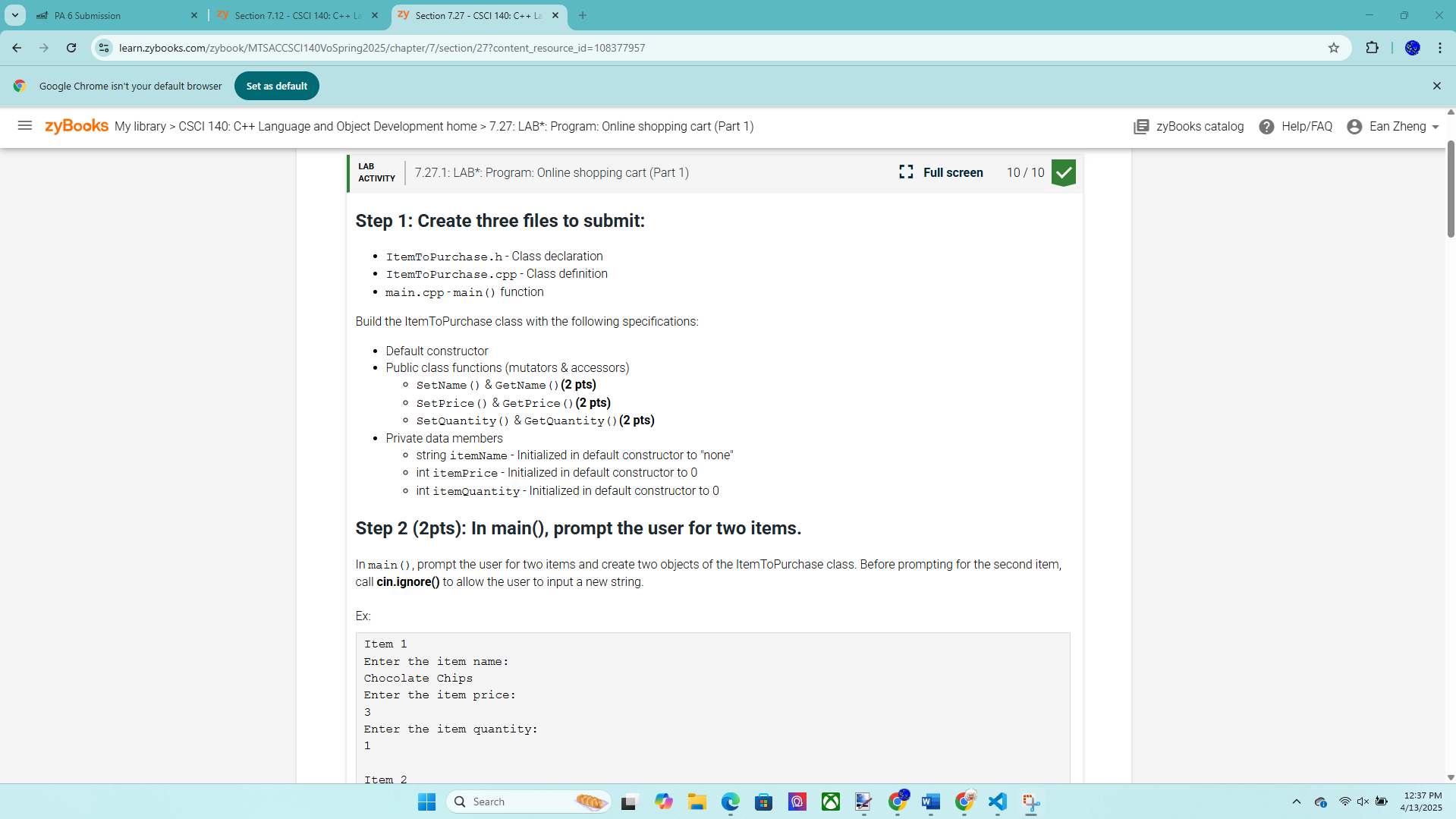
## Name(s): Ean Zheng

Exercise 1 – 7.22 LAB: Triangle area comparison

Exercise 2 – 7.23 LAB: Car value



Exercise 3 – 7.27 LAB\*: Warm up: Online shopping cart (Part 1)



Exercise 4 – Height class version 1 – more points for this exercise  
Create a class called Height and then write a driver program to test your class by creating  
some objects and performing various operations. You will continue with this class in  
future PAs so try to complete it and ask for help if needed! Your program must have at  
least three files: a Height header file (Height.h), a Height implementation file  
(Height.cpp), and an application file (HeightApp.cpp). The class has only two int data  
members feet and inches. The feet must be greater or equal to 0 (non-negative value) and  
the inches must be between 0 and 11 so validation is needed for applicable  
operations. Provide the following public member functions:  
• Constructor; must verify that feet is greater than or equal to 0 and default to 0 if  
needed; must verify that inches are between 0 and 11 and default to 0 if needed.  
o Height(int f, int i);  
• Set the feet (must verify that the value is greater than or equal to 0 and keep  
current feet and ignore bad data if applicable).  
o void setFeet(int f);  
• Set the inches (must verify that the value is between 0 and 11 and keep current  
inches and ignore bad data if applicable).  
o void setInches(int i);  
• Return the feet.  
o int getFeet() const;  
• Return the inches.  
o int getInches() const;  
• Print the height in the following format (like 5’ 6”).

o void print() const;  
• Increment the inches by one inch more (don't forget to adjust the inches and  
feet if needed).  
o void increment();  
You must try at least the following in your driver and add code to label height and new  
line for formatting as needed:  
// Create some Height objects  
Height h3(5, 8); // feet: 5, inches: 8  
Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)  
Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)  
// Print height h3  
cout << “h3: “;  
h3.print(); // h3: 5’ 8”  
cout << endl;  
// Add more code below to print h4 and h5 like h3 above  
// Perform various operations  
h3.setFeet(-2); // feet: 5, inches: 8, feet stay the same  
h3.setInches(10); // feet: 5, inches: 10  
cout << “feet: “ << h3.getFeet() << “, inches: “ << h3.getInches() << endl; // 5 10  
h4.setFeet(6); // feet: 6, inches: 5  
h4.setInches(12); // feet: 6, inches: 5, inches stay the same  
cout << “feet: “ << h4.getFeet() << “, inches: “ << h4.getInches() << endl; // 6 5  
h5.setInches(10); // feet: 6, inches: 10  
h5.increment(); // feet: 6, inches: 11  
h5.increment(); // feet: 7, inches: 0  
cout << “h3: “;  
h3.print(); // h3: 7’ 0”  
cout << endl;  
// Add more test cases if needed

Source code below:

HeightApp.cpp:

/\* Program: Height Class

Author: Ean Zheng

Class : CSCI 140

Date : 4 / 13 / 2025

Description :

I certify that the code below is my own work.

Exception(s) : N / A

\*/

#include <iostream>

using namespace std;

#include "Height.h"

int main() {

// Create some Height objects

Height h3(5, 8); // feet: 5, inches: 8

Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)

Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)

// Print height h3

cout << "h3: ";

h3.print(); // h3: 5’ 8”

cout << endl;

// Add more code below to print h4 and h5 like h3 above

// Perform various operations

h3.setFeet(-2); // feet: 5, inches: 8, feet stay the same

h3.setInches(10); // feet: 5, inches: 10

cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 5 10

h4.setFeet(6); // feet: 6, inches: 5

h4.setInches(12); // feet: 6, inches: 5, inches stay the same

cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 6 5

h5.setInches(10); // feet: 6, inches: 10

h5.increment(); // feet: 6, inches: 11

h5.increment(); // feet: 7, inches: 0

cout << "h5: ";

h5.print(); // h3: 7’ 0”

cout << endl;

//Add more test cases if needed

}

Height.cpp:

#include "Height.h"

/\* Program: Height Class

Author: Ean Zheng

Class: CSCI 140

Date: 4/13/2025

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

#include <iostream>

using namespace std;

#include "Height.h"

Height::Height(int f, int i) : feet(f), inches(i) {

if (feet < 0)feet = 0;

if (inches < 0 || inches > 11) inches = 0;

}

void Height::setFeet(int f) {

if (f >= 0)feet = f;

}

void Height::setInches(int i) {

if (i >= 0 && i <= 11) inches = i;

}

int Height::getFeet() const {

return feet;

}

int Height::getInches() const {

return inches;

}

void Height::print() const {

cout << feet << "' " << inches << "\"";

}

void Height::increment() {

++inches;

if (inches == 12) {

inches = 0;

++feet;

}

}

Height.h:

/\* Program: Height Class

Author: Ean Zheng

Class: CSCI 140

Date: 4/13/2025

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

#ifndef HEIGHT\_H

#define HEIGHT\_H

#include <string>

using namespace std;

class Height {

private:

int feet;

int inches;

public:

Height(int f = 0, int i = 0);

void setFeet(int f = 0);

void setInches(int i = 0);

int getFeet() const;

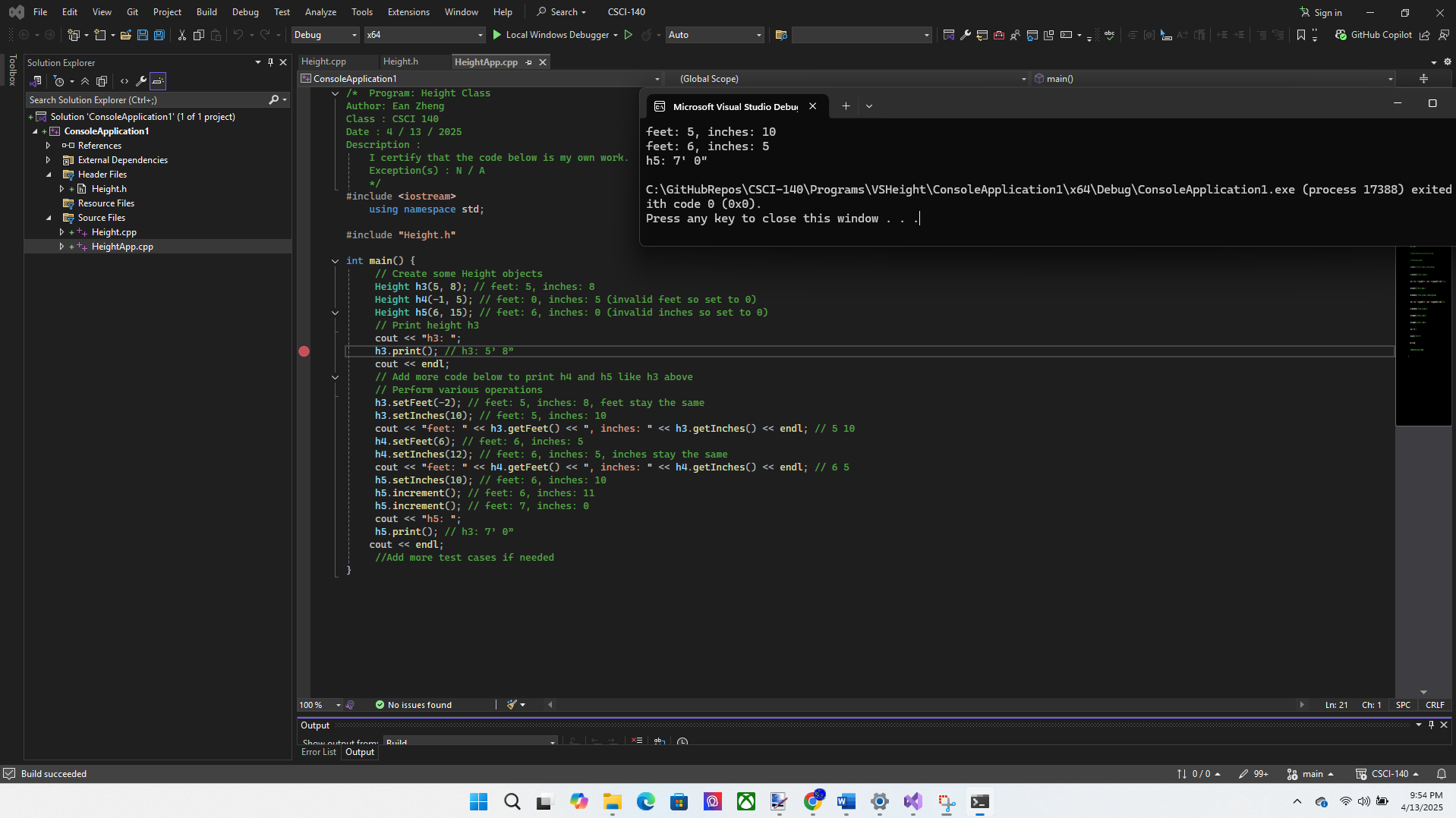
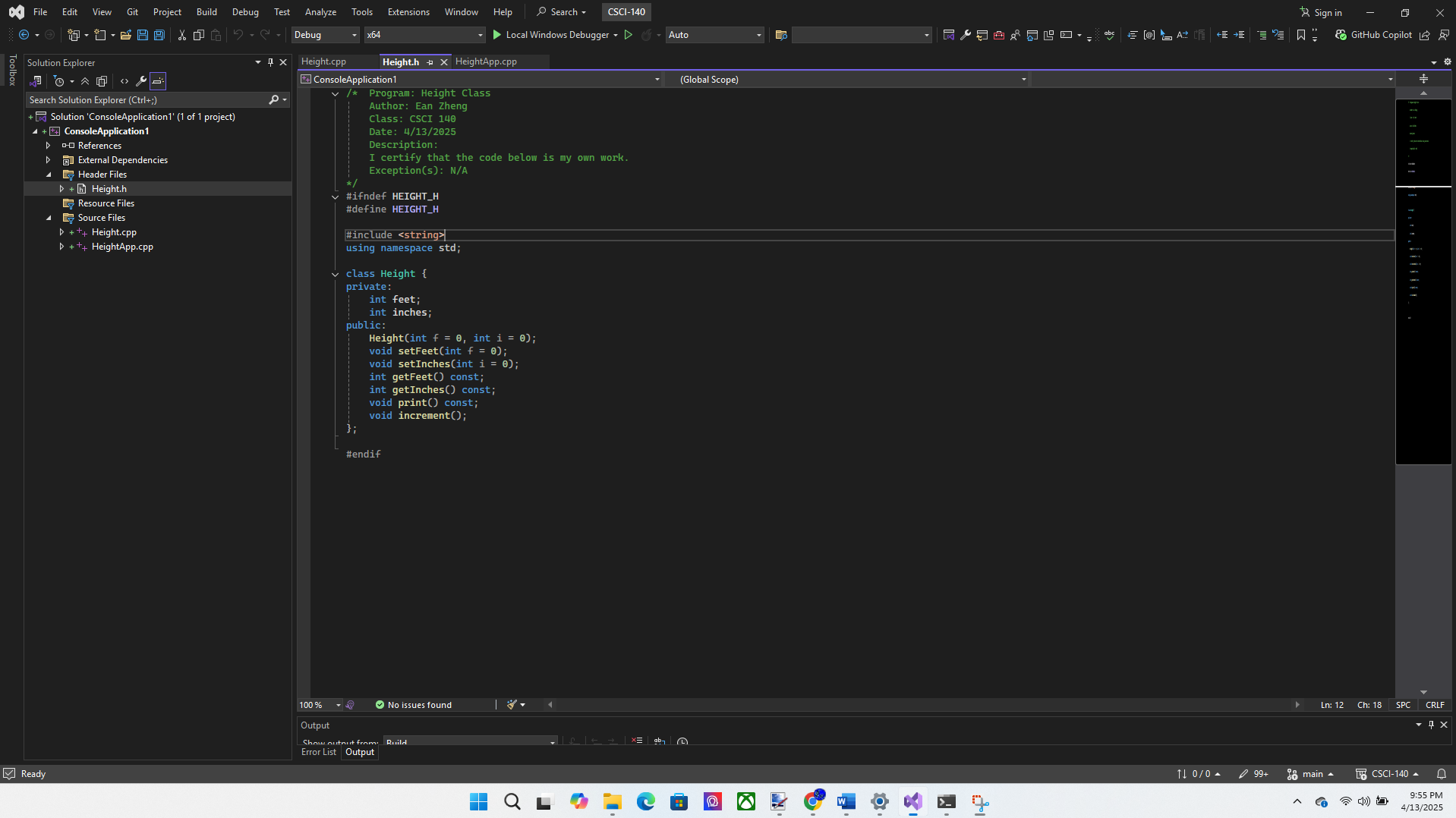
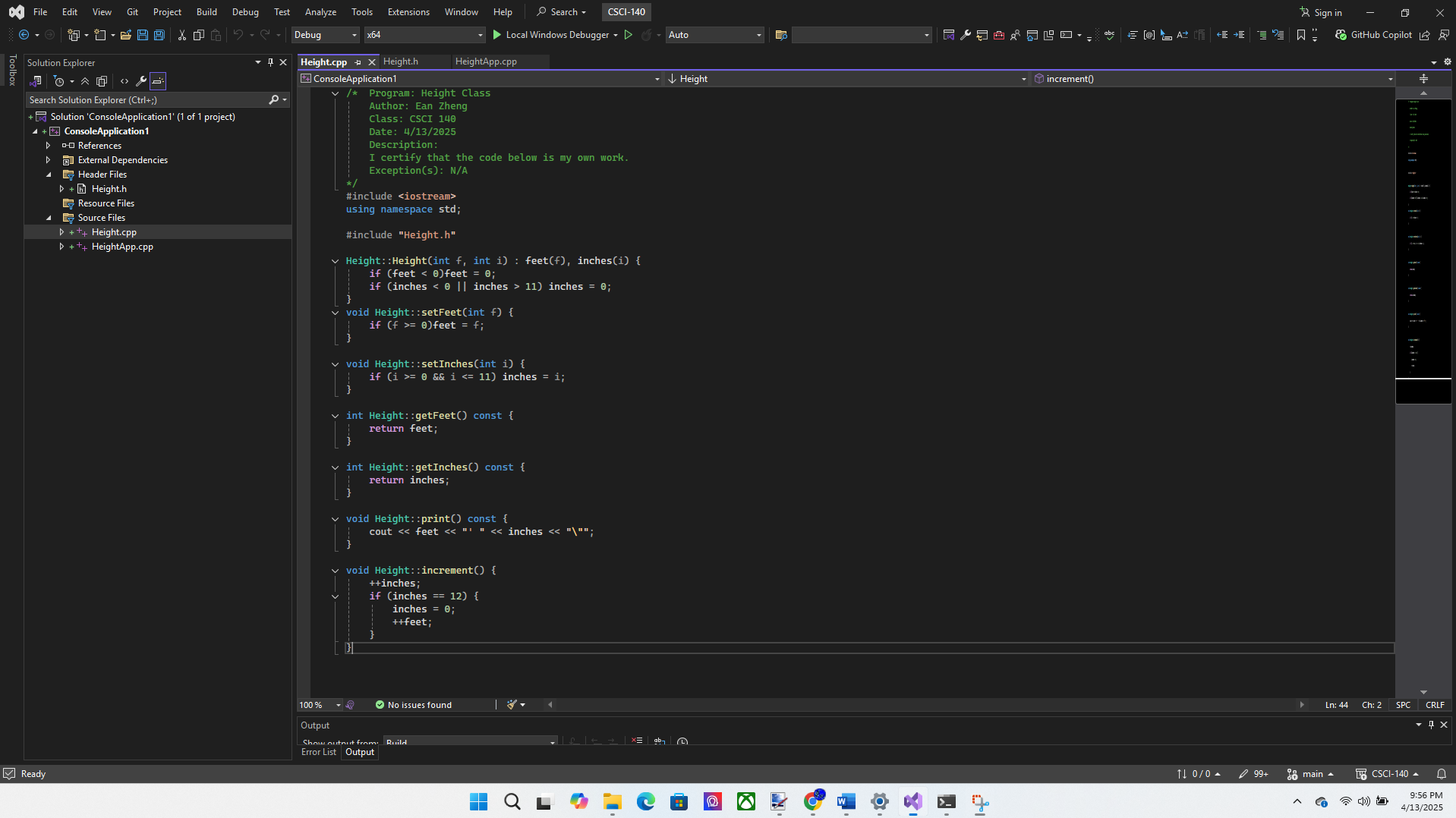
int getInches() const;

void print() const;

void increment();

};

#endif

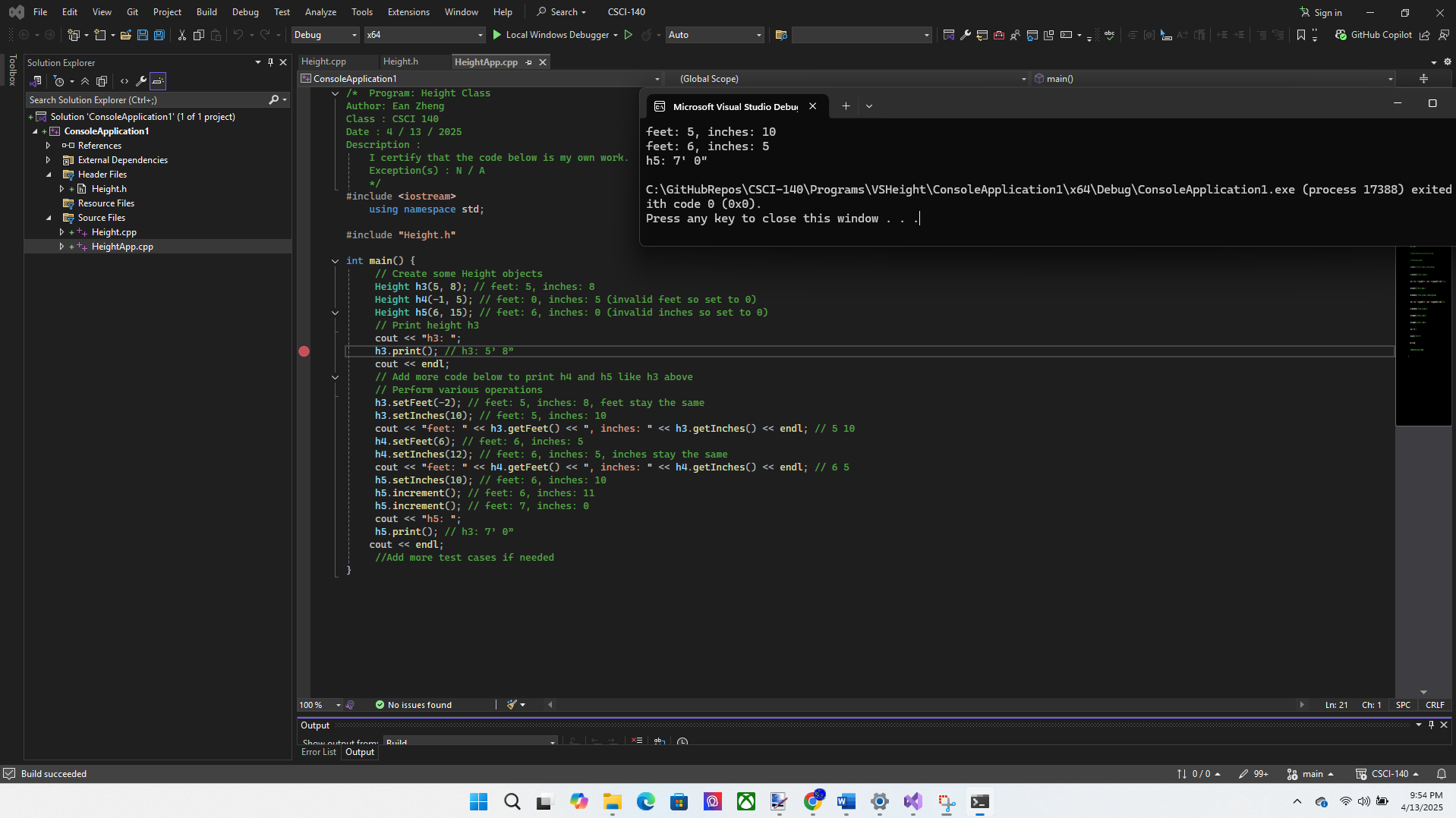
Input/output below:

h3: 5' 8"

feet: 5, inches: 10

feet: 6, inches: 5

h5: 7' 0"



Question 1: Struct originates from C language. What are some good reasons for  
grouping related data with a struct? What is the main difference between a struct in C++  
(a bit different than a struct in C) and a class?

To reduce lines of code, better organization, and reusability. A struct can only hold data values while a class can have functions and operations and also have both public and private values/functions.

Question 2: What is a default constructor? Why would you want to overload  
constructors for a class?

A special class function that initializes data members with default values upon declaration of a class object, and in the case of no arguments being given to the call. Overloading constructors allows class initialization to tolerate multiple input field cases and prevent error.

Extra Credit (2 points): Modify Height class to set up constructor(s) so the following  
Height objects can be created in the driver. Set up a Height array as specified below and  
print the heights in the driver as well (can modify regular version and submit one  
version).  
// Create 2 Height objects  
Height h1; // feet: 0, inches: 0  
Height h2(5); // feet: 5, inches: 0  
// Create some Height objects (same as original version)

Height h3(5, 8); // feet: 5, inches: 8  
Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)  
Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)  
// Set up an array to hold 5 different heights above and use a loop to print the heights  
// Same code as exercise 4 below

Source code below:

HeightEC.cpp:

/\* Program: Height Class

Author: Ean Zheng

Class : CSCI 140

Date : 4 / 13 / 2025

Description :

I certify that the code below is my own work.

Exception(s) : N / A

\*/

#include <iostream>

using namespace std;

#include "Height.h"

int main() {

// Create 2 Height objects

Height h1; // feet: 0, inches: 0

Height h2(5); // feet: 5, inches: 0

// Create some Height objects (same as original version)

Height h3(5, 8); // feet: 5, inches: 8

Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)

Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)

// Print height h3

Height array[] {h1, h2, h3, h4, h5};

for (int i = 0; i < 5; ++i) {

array[i].print();

cout << endl;

}

cout << "h3: ";

h3.print(); // h3: 5’ 8”

cout << endl;

// Add more code below to print h4 and h5 like h3 above

// Perform various operations

h3.setFeet(-2); // feet: 5, inches: 8, feet stay the same

h3.setInches(10); // feet: 5, inches: 10

cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 5 10

h4.setFeet(6); // feet: 6, inches: 5

h4.setInches(12); // feet: 6, inches: 5, inches stay the same

cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 6 5

h5.setInches(10); // feet: 6, inches: 10

h5.increment(); // feet: 6, inches: 11

h5.increment(); // feet: 7, inches: 0

cout << "h5: ";

h5.print(); // h3: 7’ 0”

cout << endl;

//Add more test cases if needed

}

Height.cpp:

/\* Program: Height Class

Author: Ean Zheng

Class: CSCI 140

Date: 4/13/2025

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

#include <iostream>

using namespace std;

#include "Height.h"

Height::Height(int f, int i) : feet(f), inches(i) {

if (feet < 0)feet = 0;

if (inches < 0 || inches > 11) inches = 0;

}

void Height::setFeet(int f) {

if (f >= 0)feet = f;

}

void Height::setInches(int i) {

if (i >= 0 && i <= 11) inches = i;

}

int Height::getFeet() const {

return feet;

}

int Height::getInches() const {

return inches;

}

void Height::print() const {

cout << feet << "' " << inches << "\"";

}

void Height::increment() {

++inches;

if (inches == 12) {

inches = 0;

++feet;

}

}

Height.h:

/\* Program: Height Class

Author: Ean Zheng

Class: CSCI 140

Date: 4/13/2025

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

#ifndef HEIGHT\_H

#define HEIGHT\_H

#include <string>

using namespace std;

class Height {

private:

int feet;

int inches;

public:

Height(int f = 0, int i = 0);

void setFeet(int f = 0);

void setInches(int i = 0);

int getFeet() const;

int getInches() const;

void print() const;

void increment();

};

#endif

Input/output below:

0' 0"

5' 0"

5' 8"

0' 5"

6' 0"

h3: 5' 8"

feet: 5, inches: 10

feet: 6, inches: 5

h5: 7' 0"

