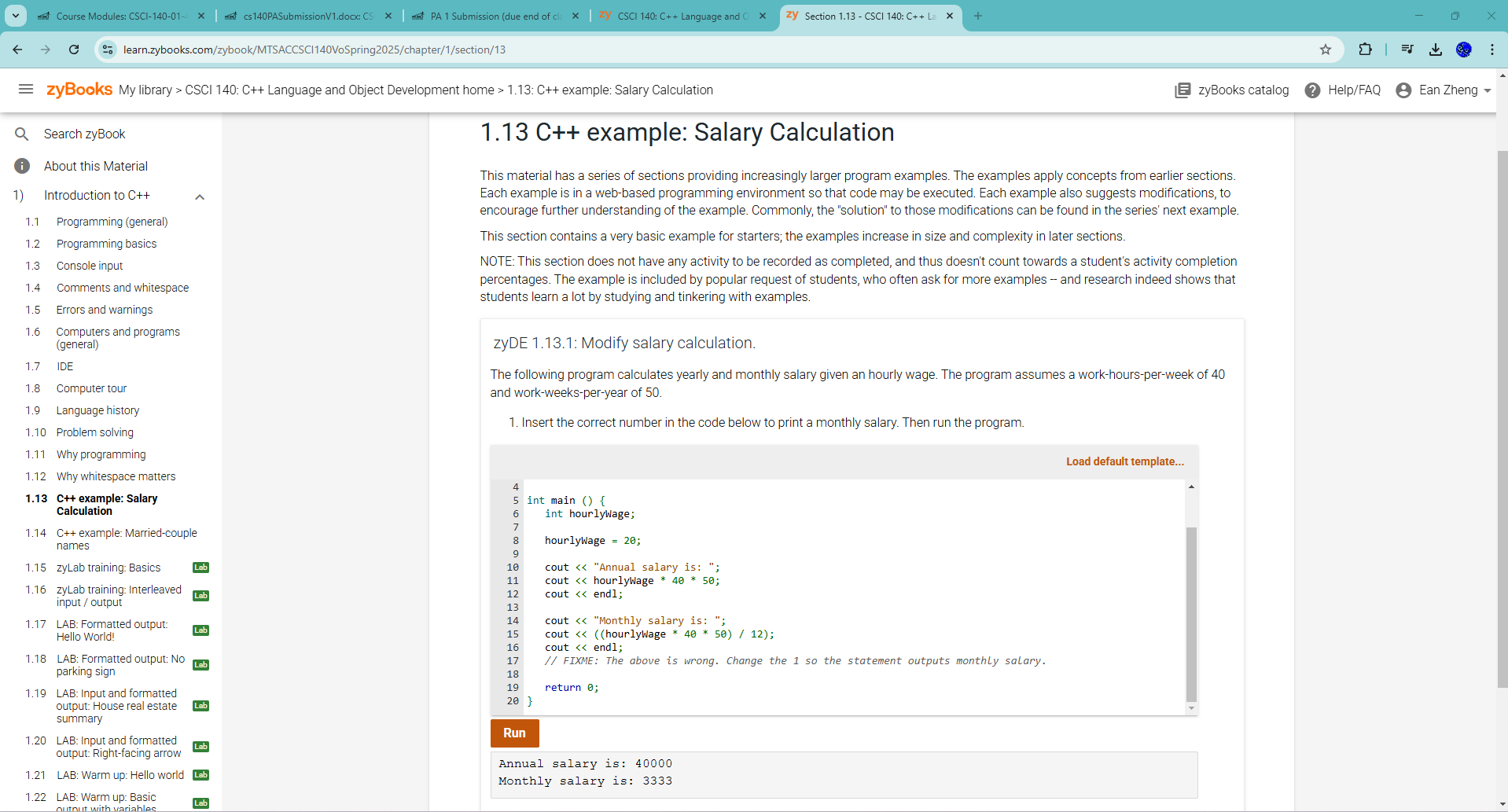
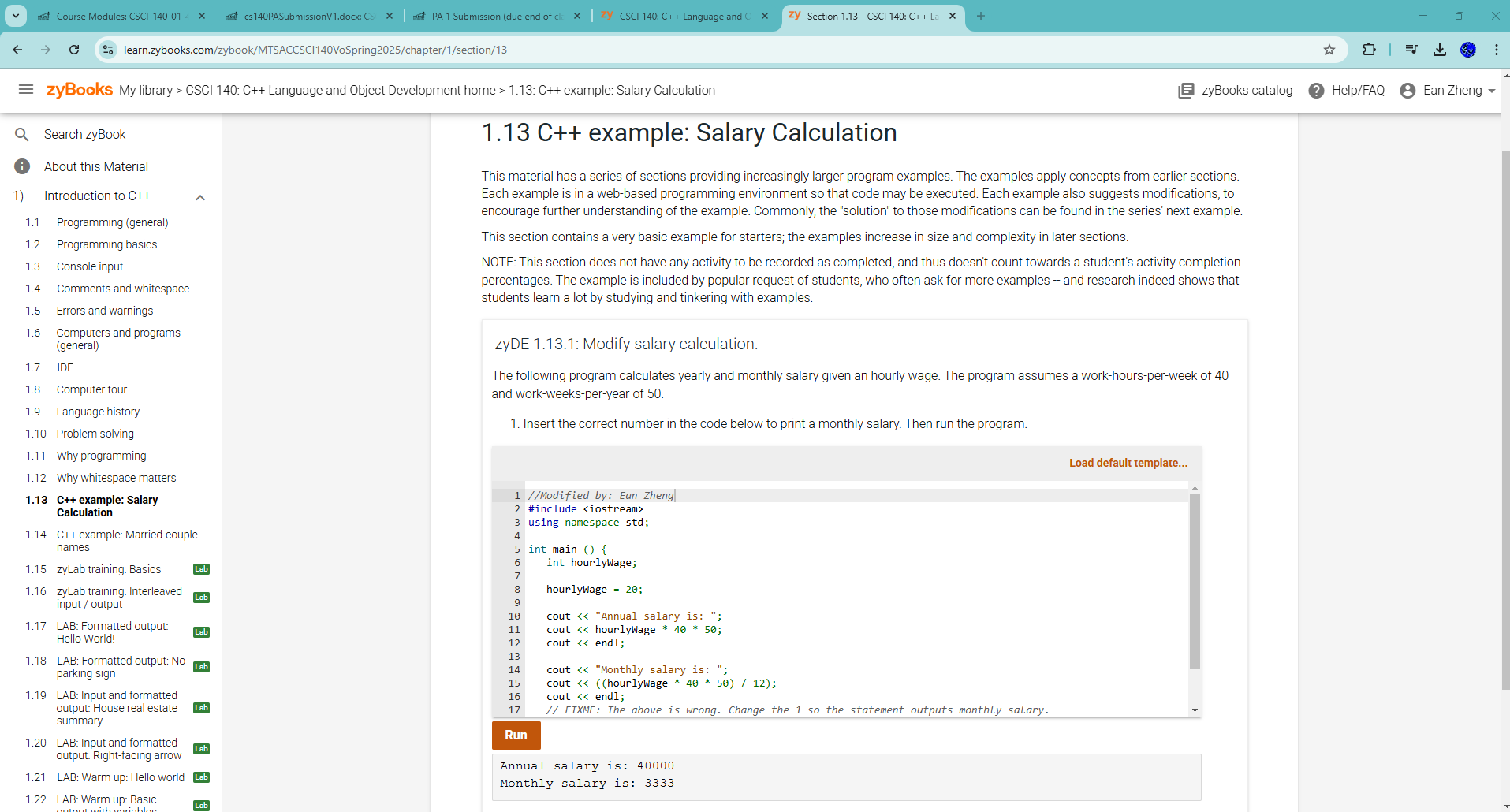
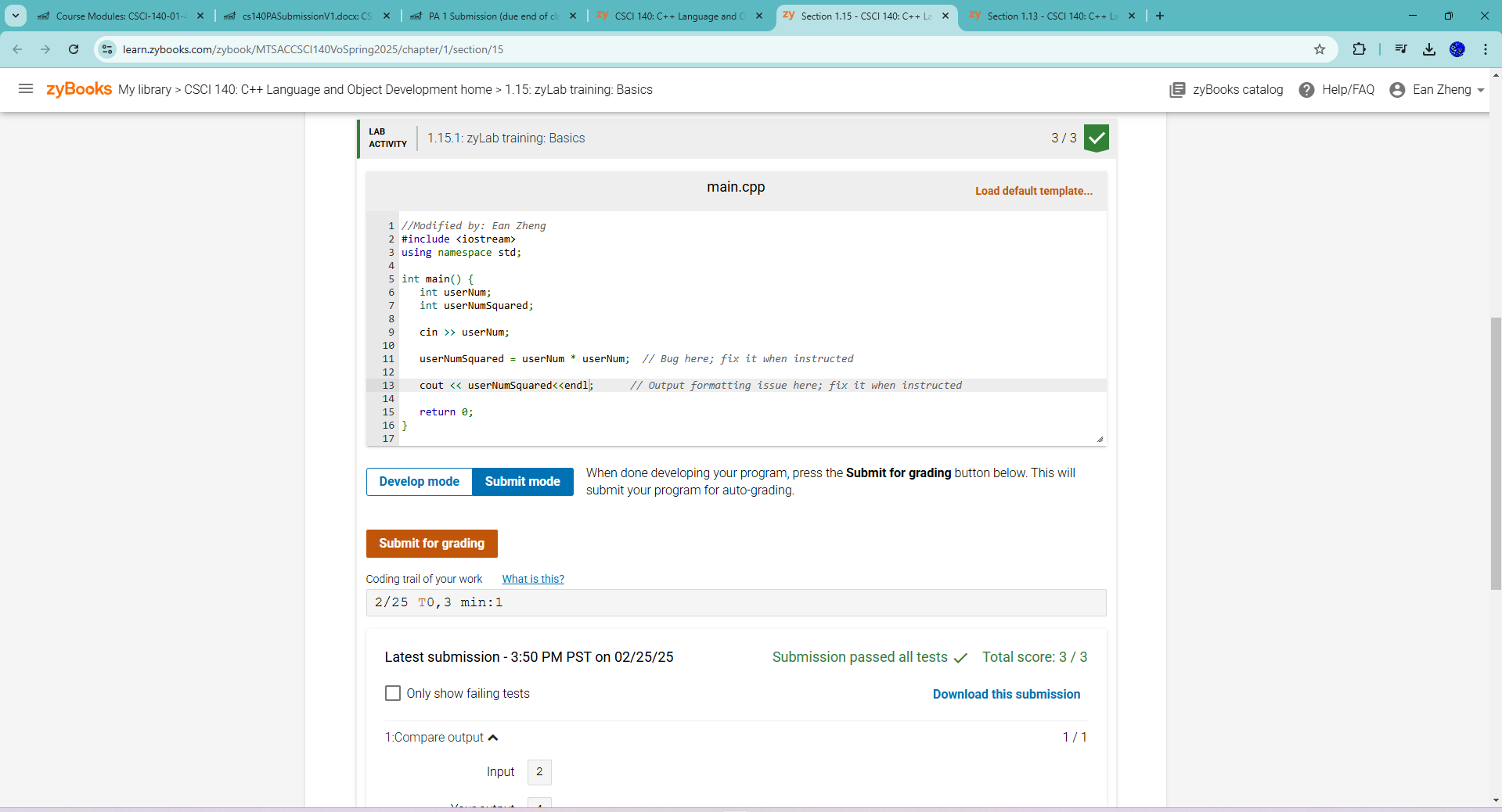
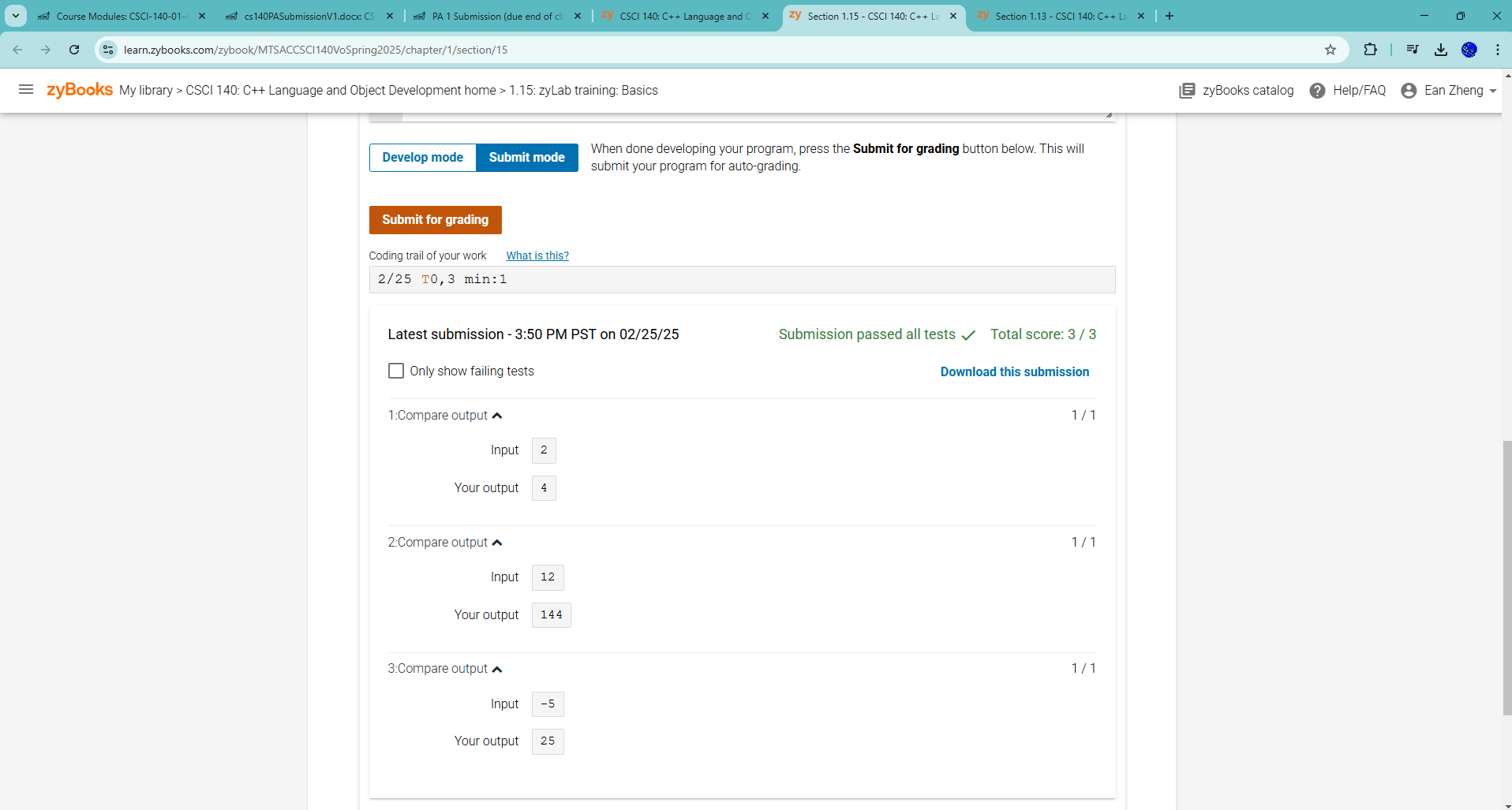
## CSCI 140 PA 1 Submission

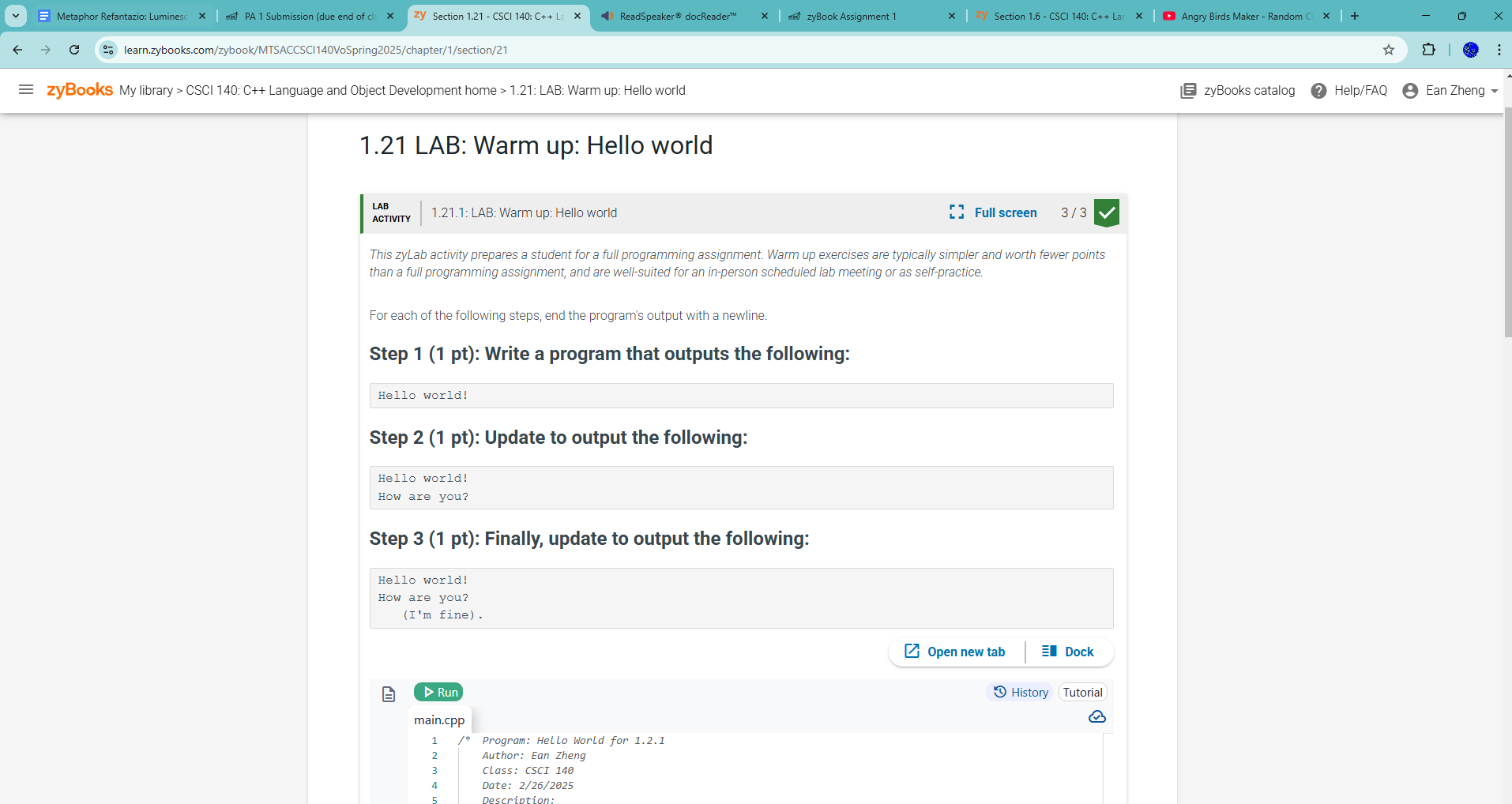
## Due Date:3/3/2025 Late (date and time):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

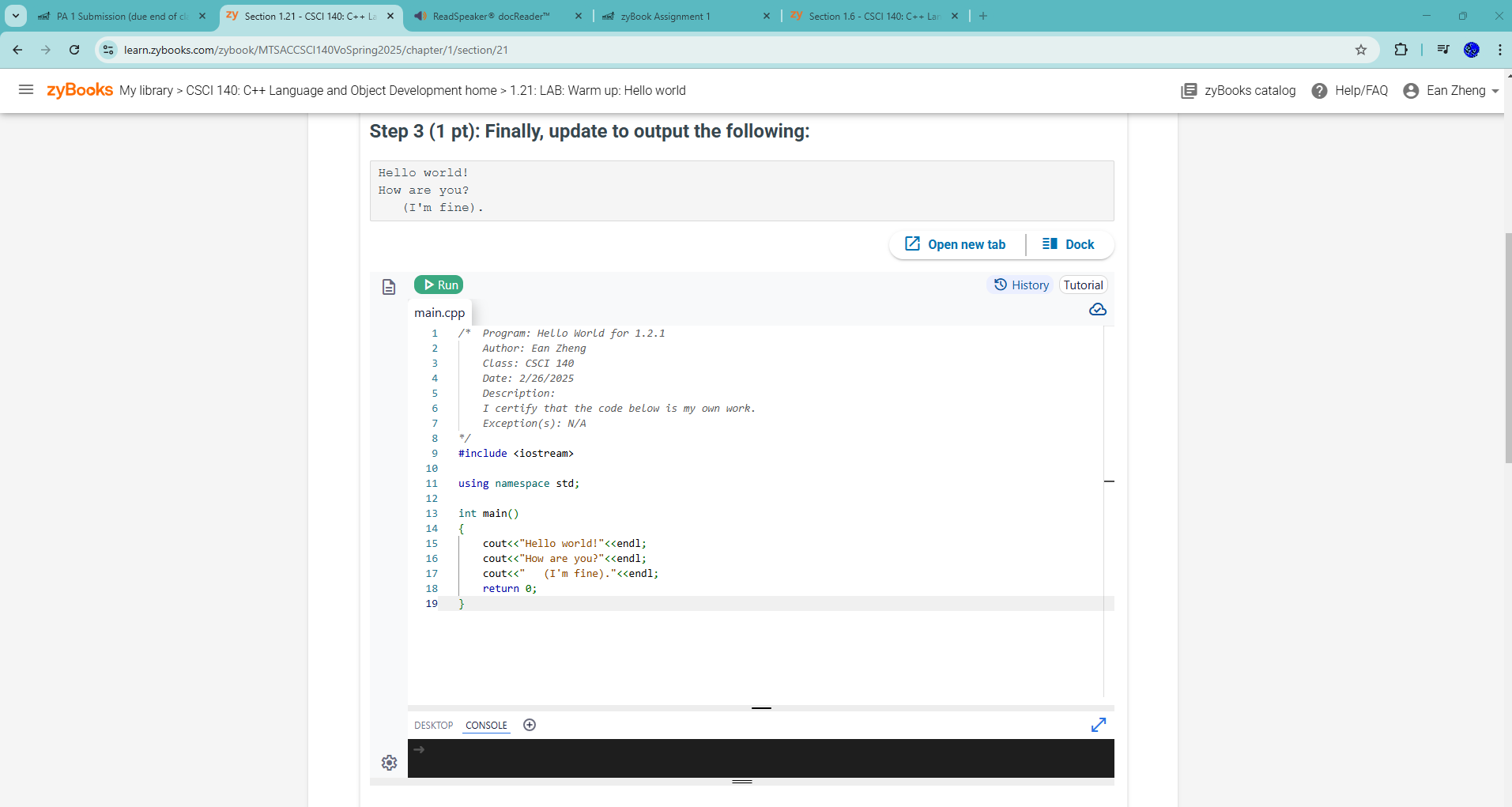
## Name: Ean Zheng

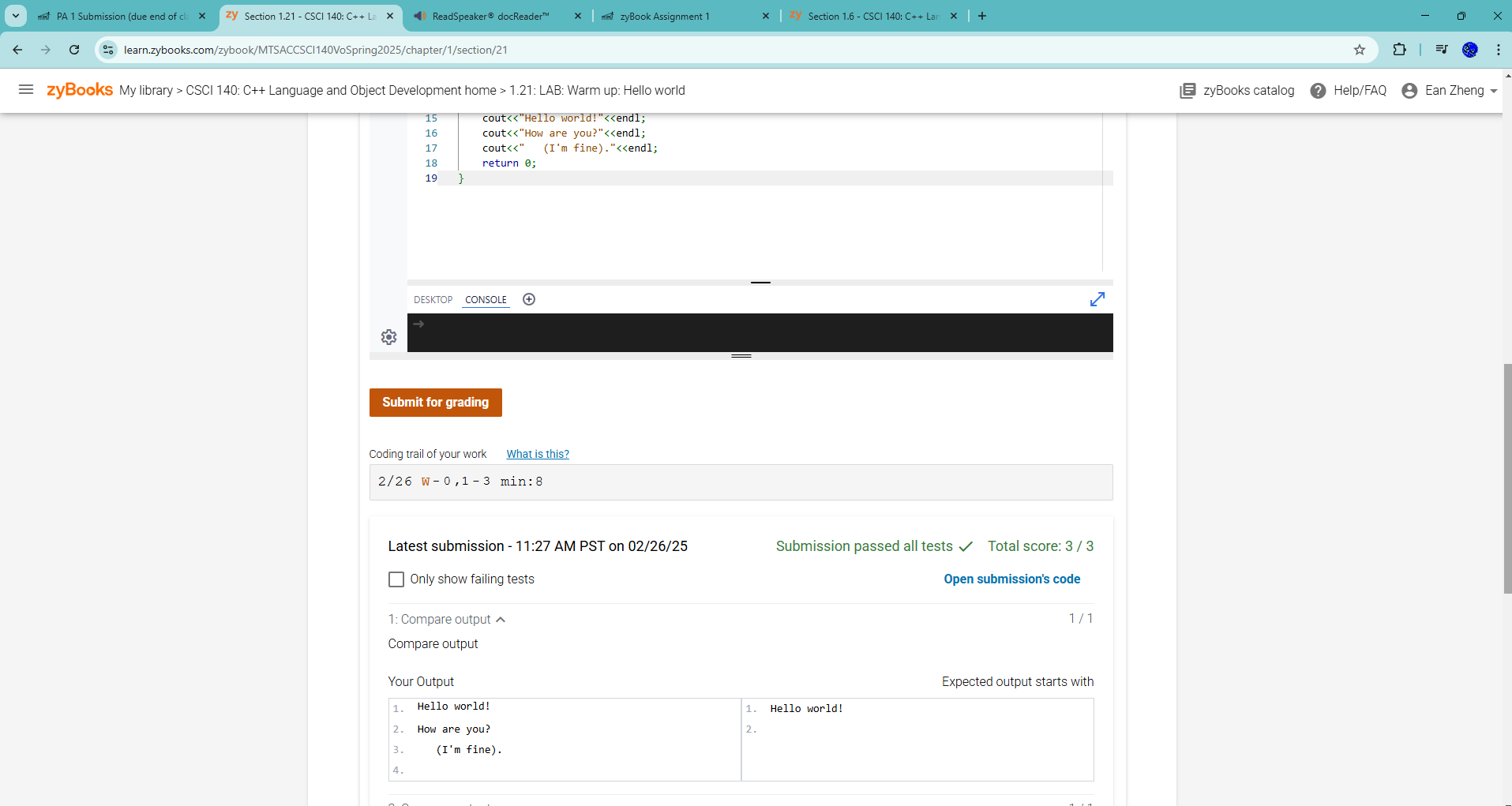
Exercise 1 – zyBook 1.13 zyLab training: Basics:

Exercise 1 – zyBook 1.15 zyLab training: Basics:**** 

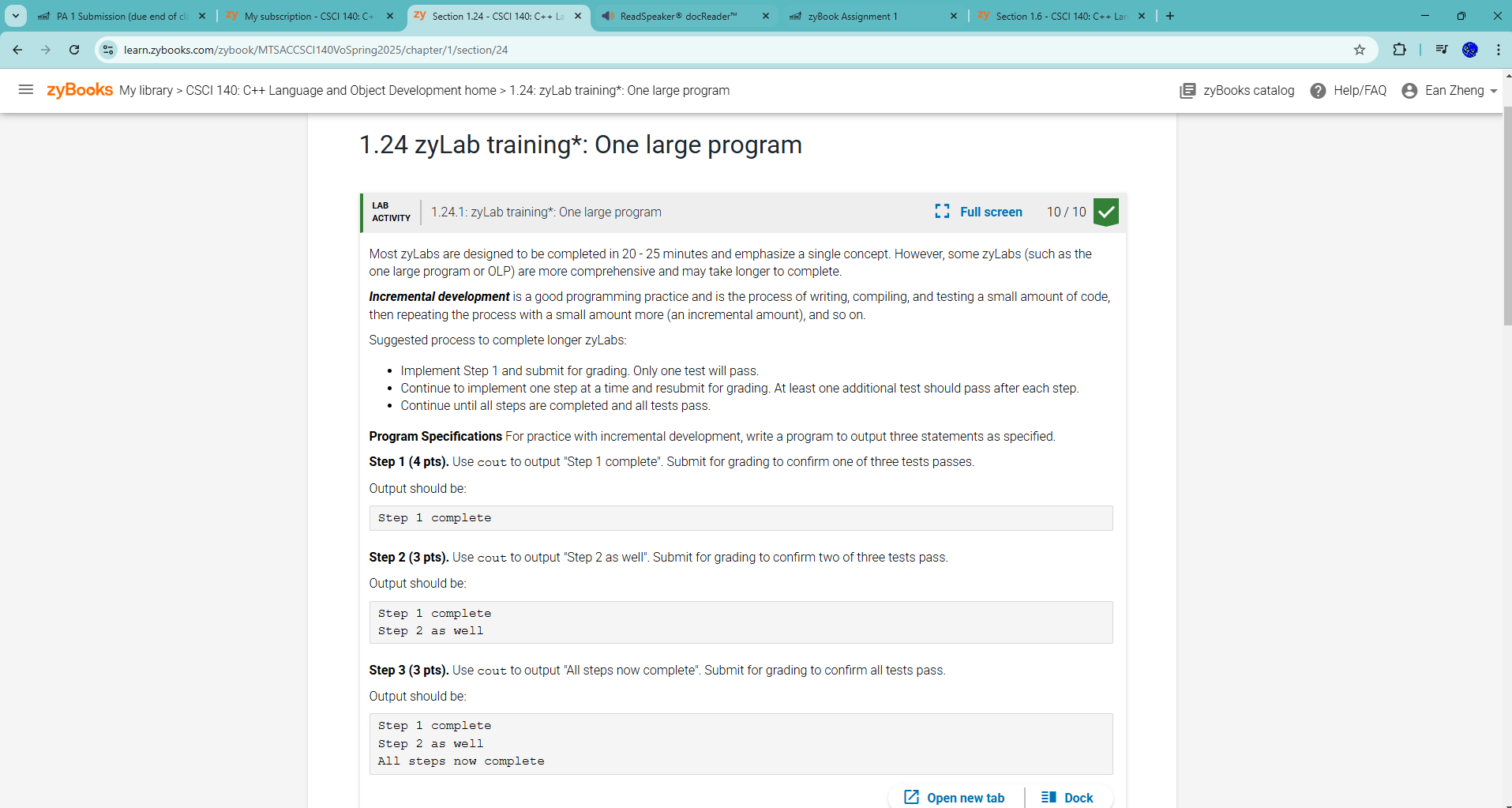
Exercise 2 – zyBook 1.21 LAB\*: Program: ASCII art:

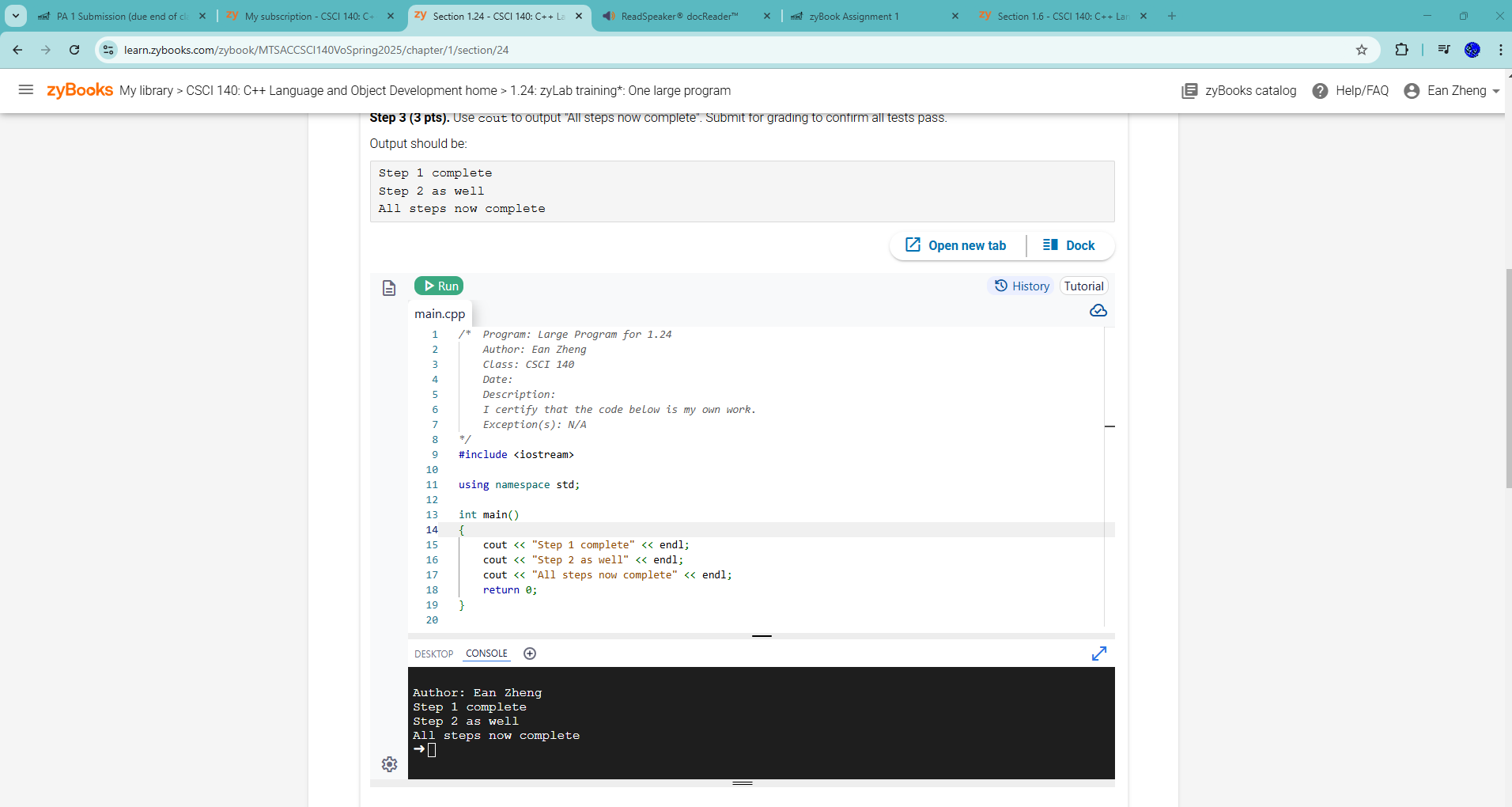


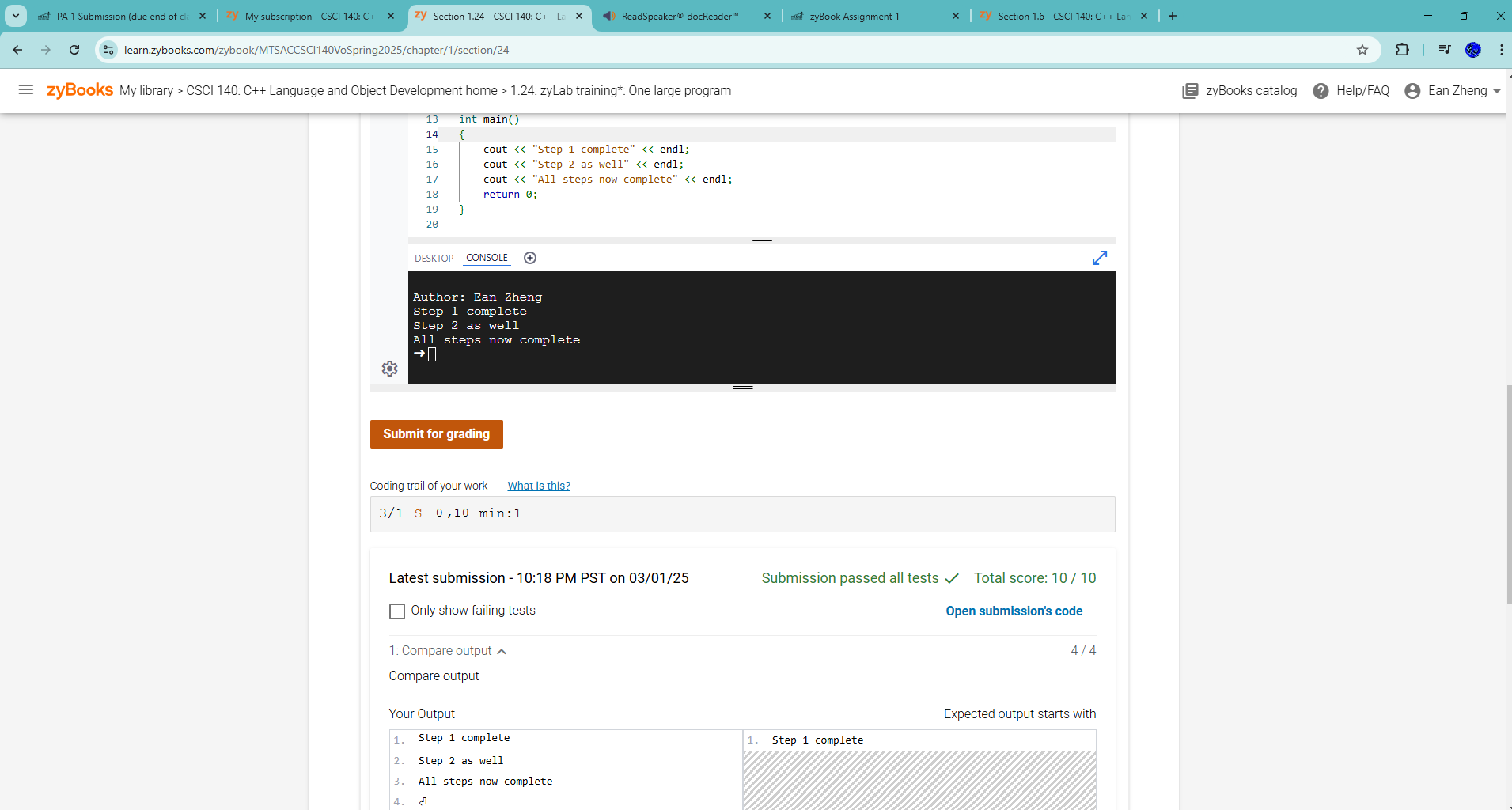




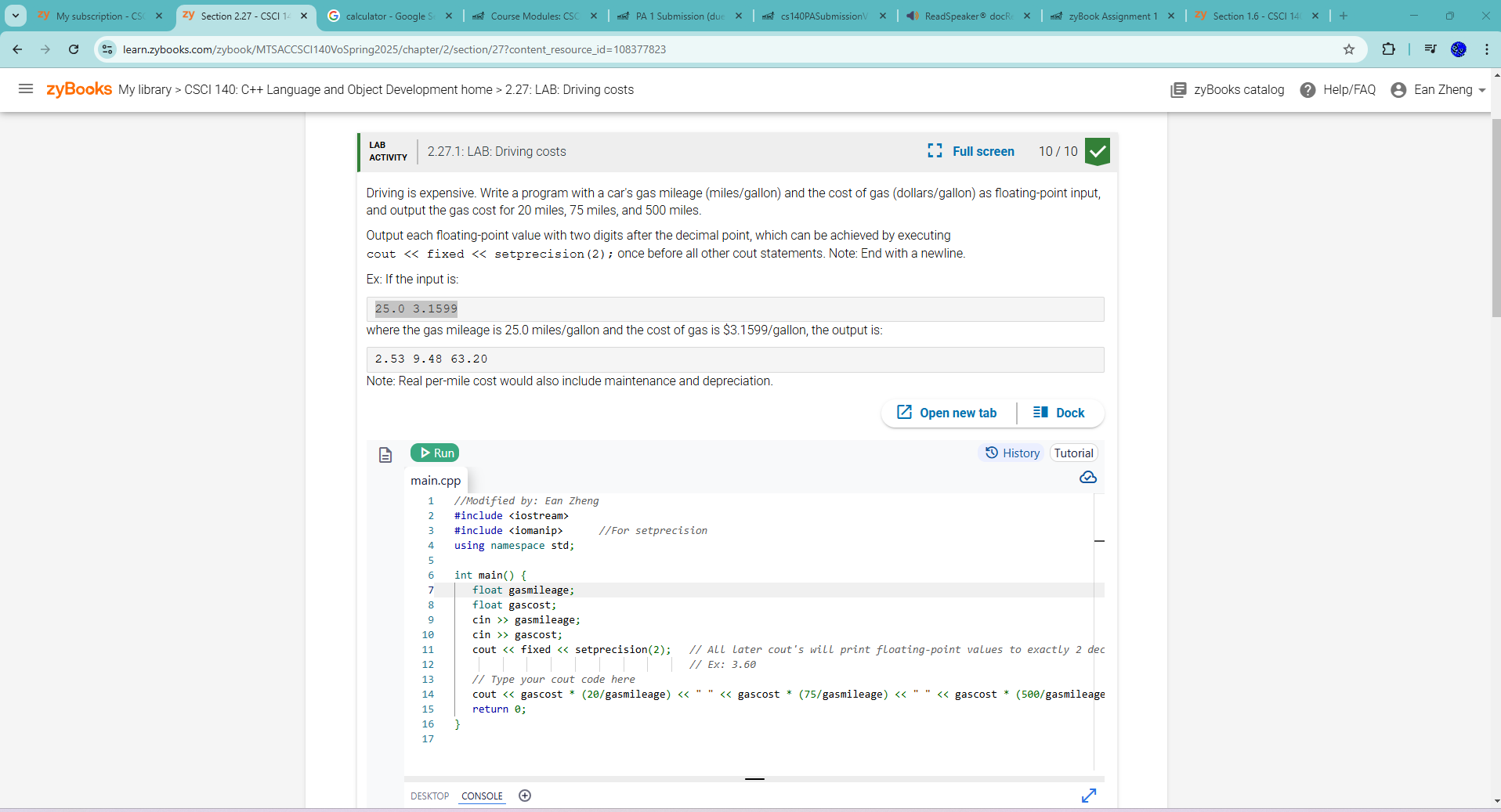
Exercise 3 – zyBook 1.24 zyLab training: One large program:

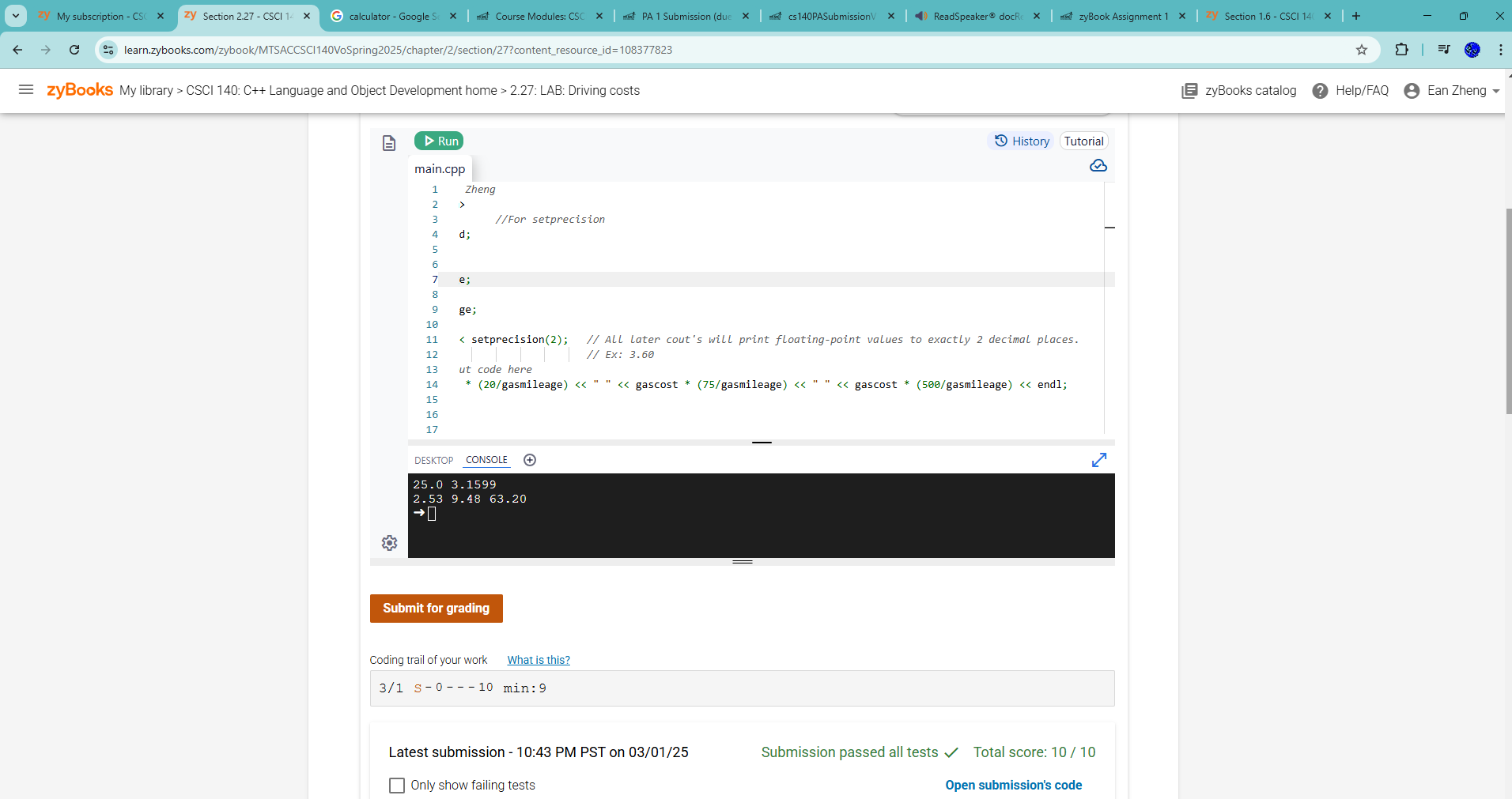




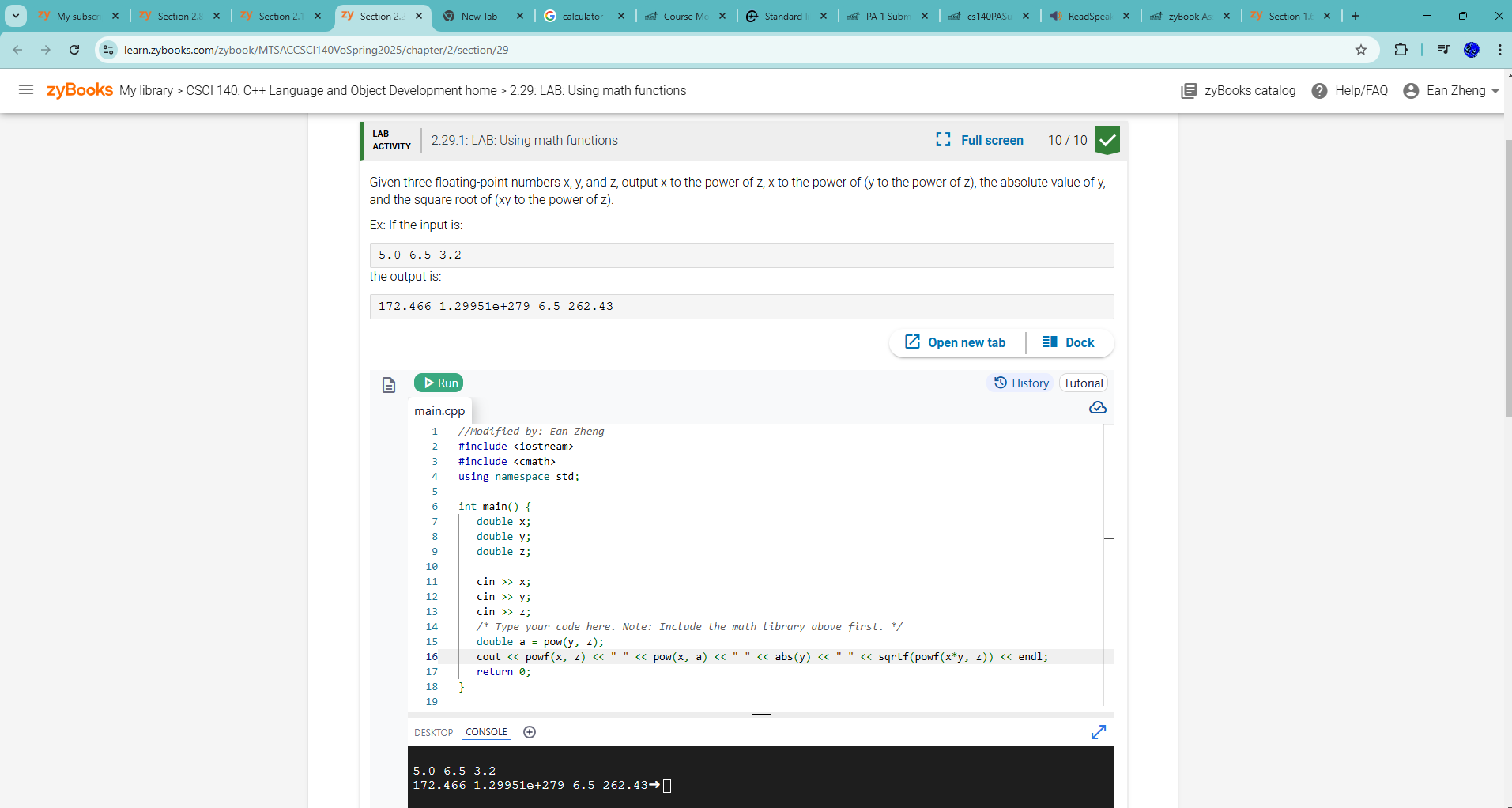


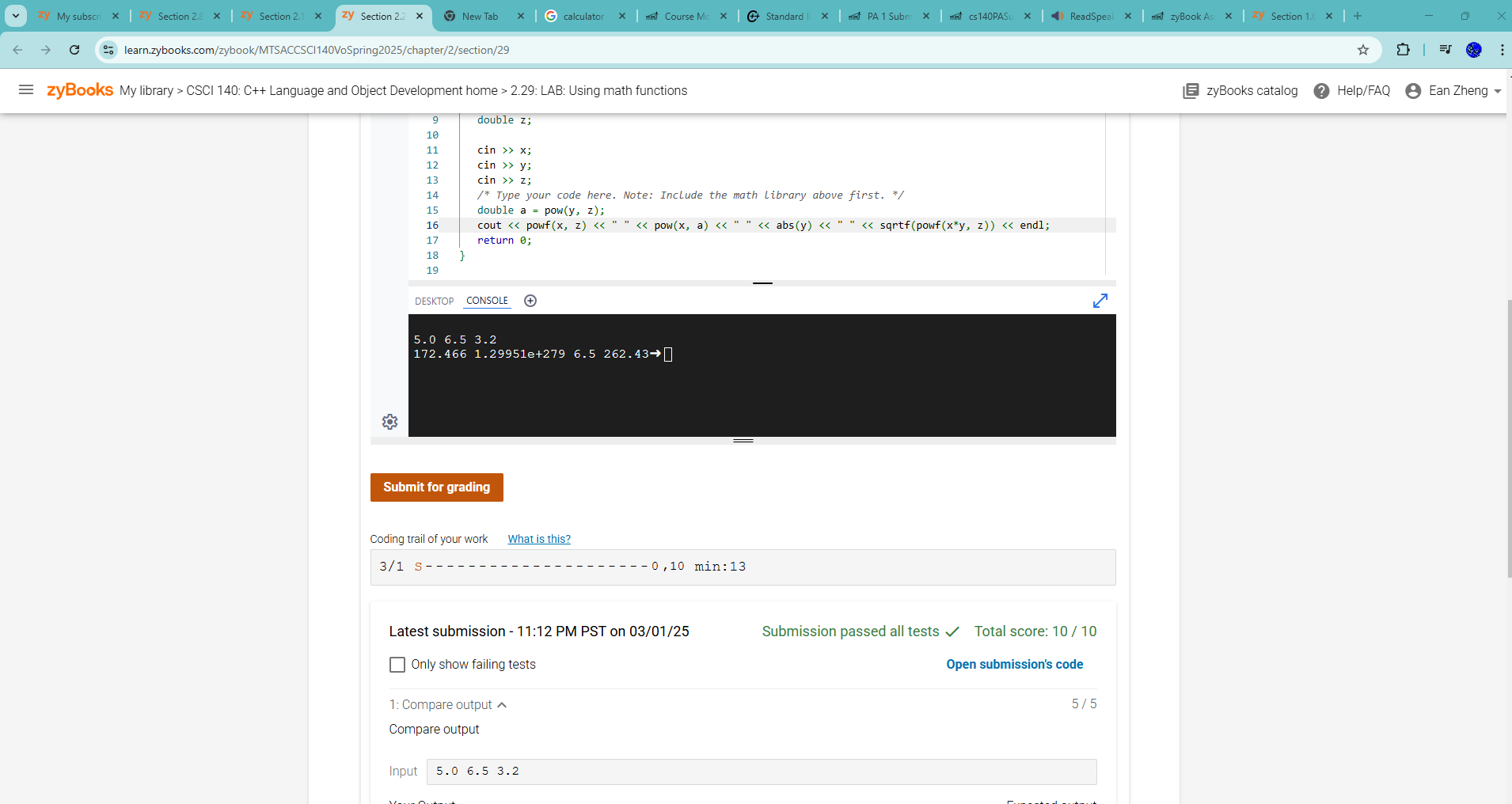
Exercise 4 – zyBook 2.27 LAB: Driving costs:



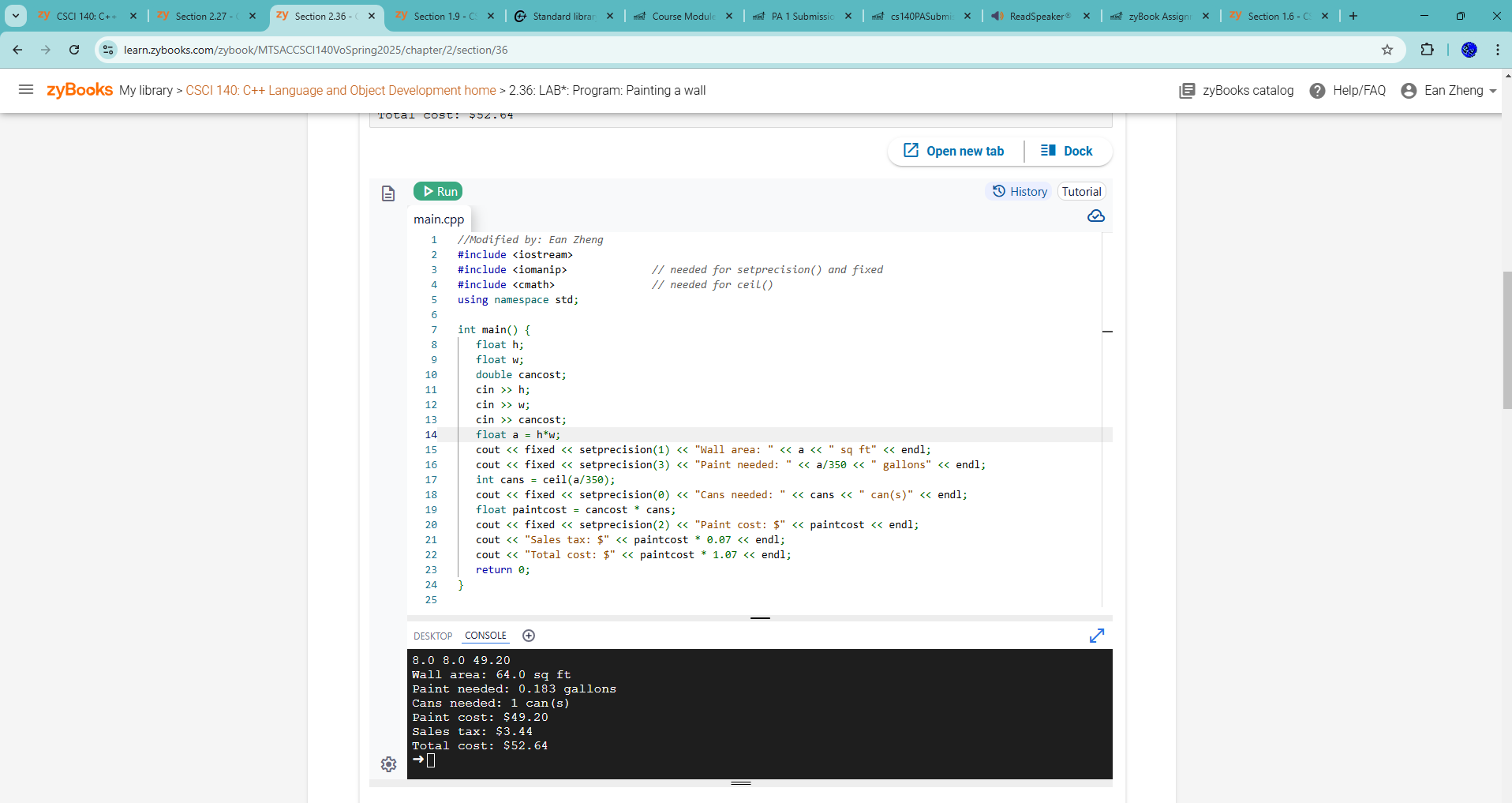


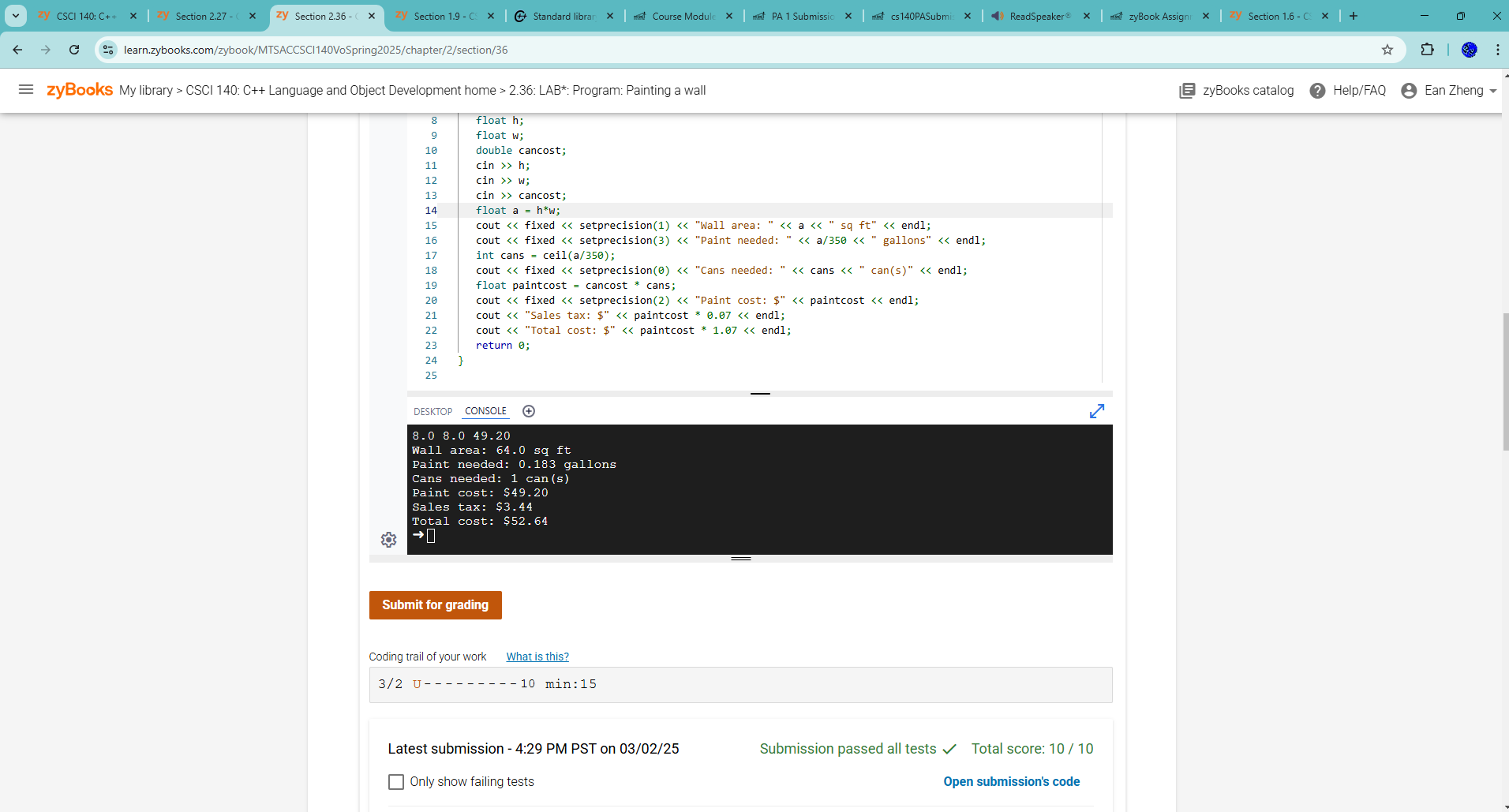
Exercise 5 – zyBook 2.29 LAB: Using math functions:





Exercise 6 – zyBook 2.36 LAB\*: Program: Painting a wall:





Exercise 7 – Writing Your First Program from Scratch  
Step 1: Develop a design that leads to an algorithm that will read in a number that  
represents the number of total seconds. The output will be the equivalent number of  
hours, number of minutes, and leftover seconds. For examples, an input of 10000  
seconds would be equivalent to 2 hours, 46 minutes, and 40 seconds. Provide  
pseudocode for this step.

Pseudocode below:

Variables: input, seconds, minutes, and hours, all int variables. Cin input variable to get input.

Divide input by 60 and floor it, store this as minutes variable, store mod of input by 60 as seconds variable

Divide minutes by 60 and floor it, store this as hours variable, update minutes variable as mod of minutes by 60

Print hours, minutes, and seconds with labels on them.

Step 2: Use your own IDE such as MS Visual Studio to develop your program. Call this  
program timeConversion.cpp. Translate pseudocode to C+ code. Compile the program.  
If you get compile errors, try to fix them, and recompile until your program is free of  
syntax errors.

Source code below:

/\*  Program: First Scratch Program for Exercise 7, PA Submission 1

    Author: Ean Zheng

    Class: CSCI 140

    Date: 3/2/2025

    Description:

    I certify that the code below is my own work.

    Exception(s): N/A

\*/

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    cout << "Author: Ean Zheng" << endl;

    int input;

    int seconds;

    int minutes;

    int hours;

    cin >> input;

    minutes = floor(input/60);

    seconds = input%60;

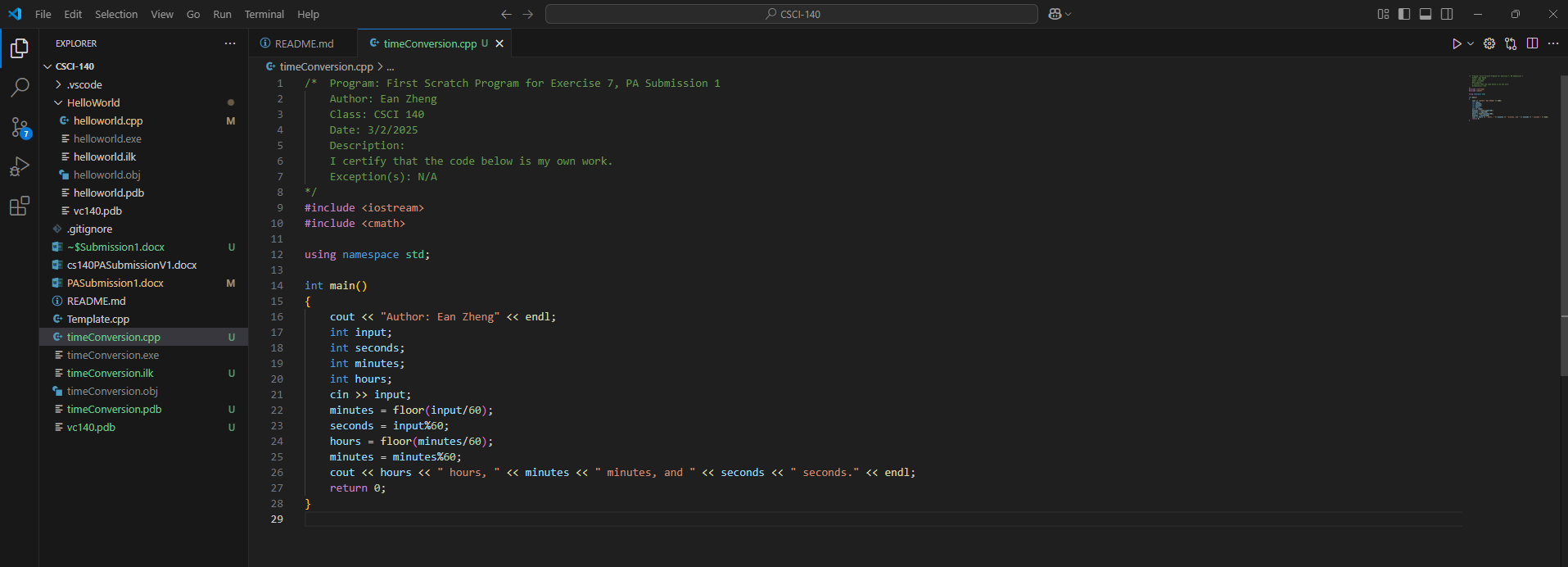
    hours = floor(minutes/60);

    minutes = minutes%60;

    cout << hours << " hours, " << minutes << " minutes, and " << seconds << " seconds." << endl;

    return 0;

}



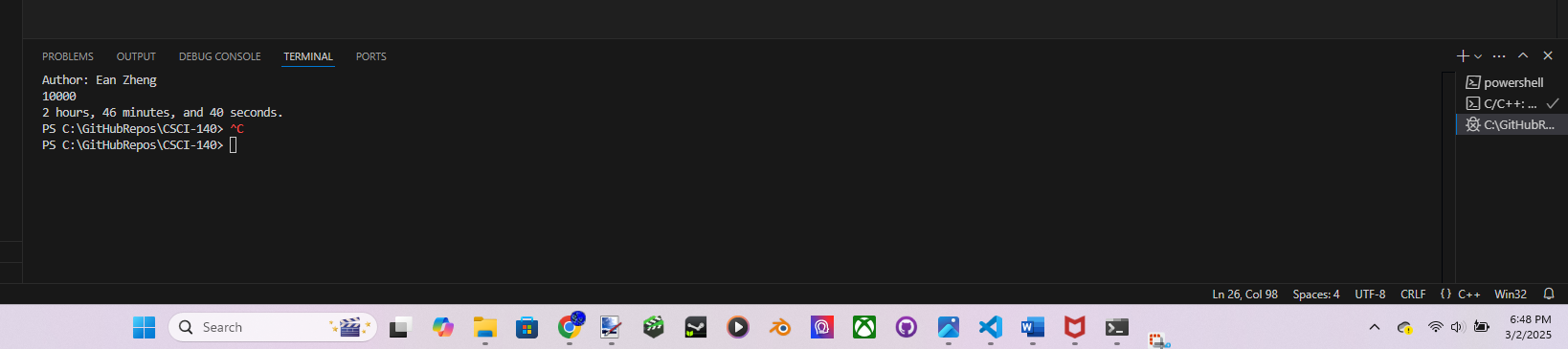
Step 3: Run the program with 10000 seconds. Is your output what you expect from the  
input you gave? If not, try to find and correct the logic error and run the program again.  
Continue this process until you have a program that produces the correct result. Try  
another value, 601 seconds, and confirm it produces the correct result as well.

Input/output below:

Author: Ean Zheng

10000

2 hours, 46 minutes, and 40 seconds.



Question 1: Explain why source code is much more important than executable code for  
a SW developer.

Because the source code is the written down code that hasn’t been compiled into executable code yet, and can still be edited or changed. Executable code is code compiled into 0 and 1s and can’t be easily edited in the same way as source code, which is before it was compiled.

Question 2: How do you know for sure that your program is working correctly?

If there are no syntax, runtime, or compilation errors, and the output is correct for every single intended scenario.

Extra Credit (2 points): Develop a design that leads to an algorithm and a program that  
will read in a positive integer that represents a potential value of 1 less than power of 2.  
The output will be “yes” for a 1 less than power of 2 and “no” when it is not. Some  
examples of yes: 0, 7, 4095. Some examples of no: 5, 10, 5000. You must start with  
pseudocode and provide pseudocode for this exercise in addition to the program and its  
output. Note: do not use bit shift operators and you may need selection, repetition, and  
possibly a function for this exercise

Pseudocode below if applicable:

One int input variable that is later put in cin.

Input variable is added 1.

While loop for while input is not equal 1.:

If input mod 2 = 0, divide input by 2

Else if input mod 2 isn’t 0, print no and return 0.

If while loop exits, print yes and return.

Source code below:

/\* Program: Extra Credit for PA 1

Author: Ean Zheng

Class: CSCI 140

Date: 3/2/2025

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

#include <iostream>

using namespace std;

int main()

{

cout << "Author: Ean Zheng" << endl;

int input;

cin >> input;

input++;

while(input != 1){

if(input%2==0){

input/=2;

}else if(input%2!=0){

cout << "no";

return 0;

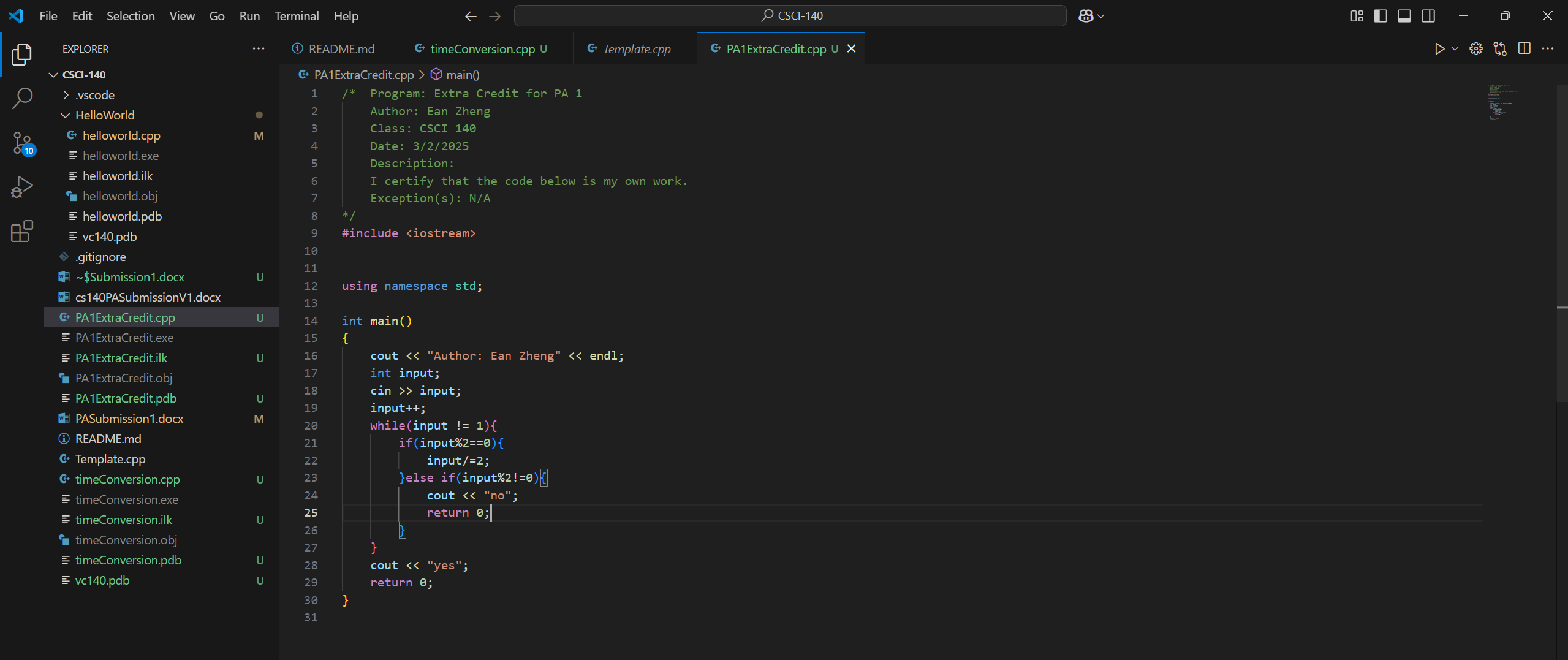
}

}

cout << "yes";

return 0;

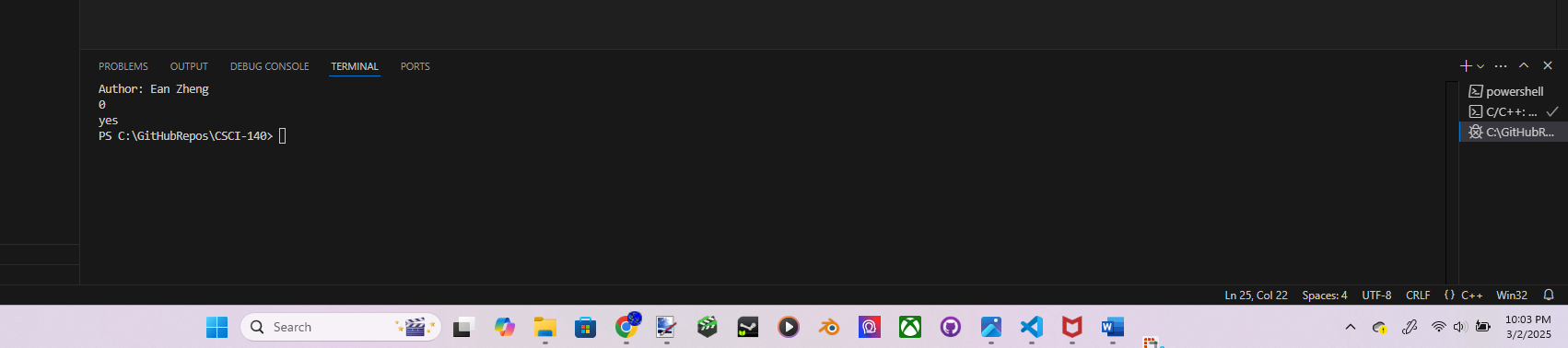
}



Input/output below:

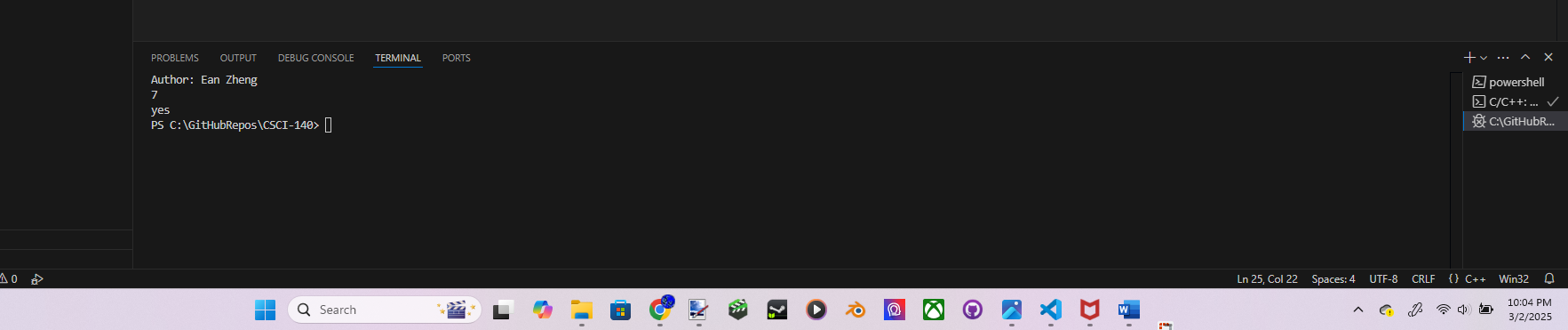
0

yes



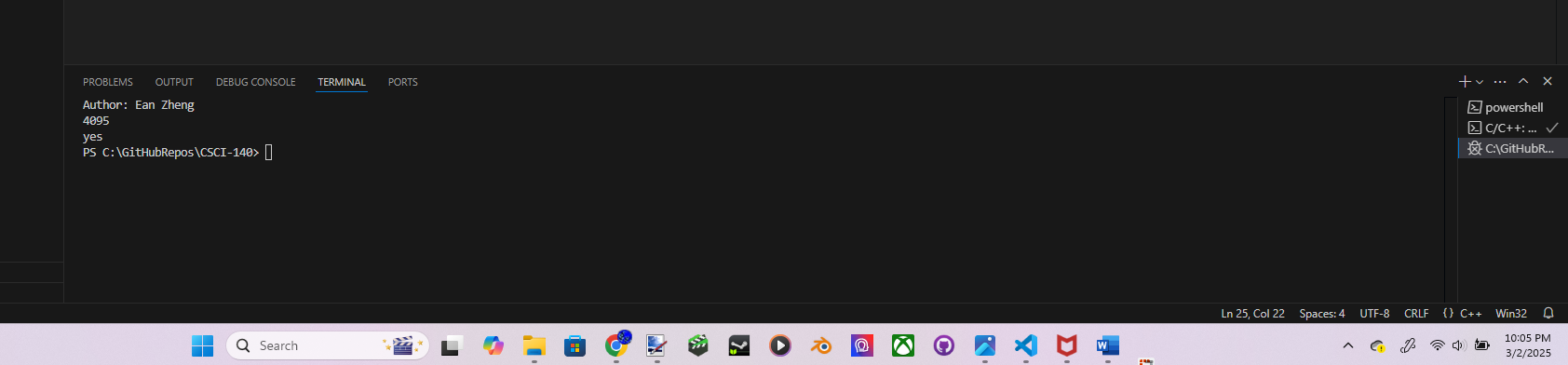
7

yes



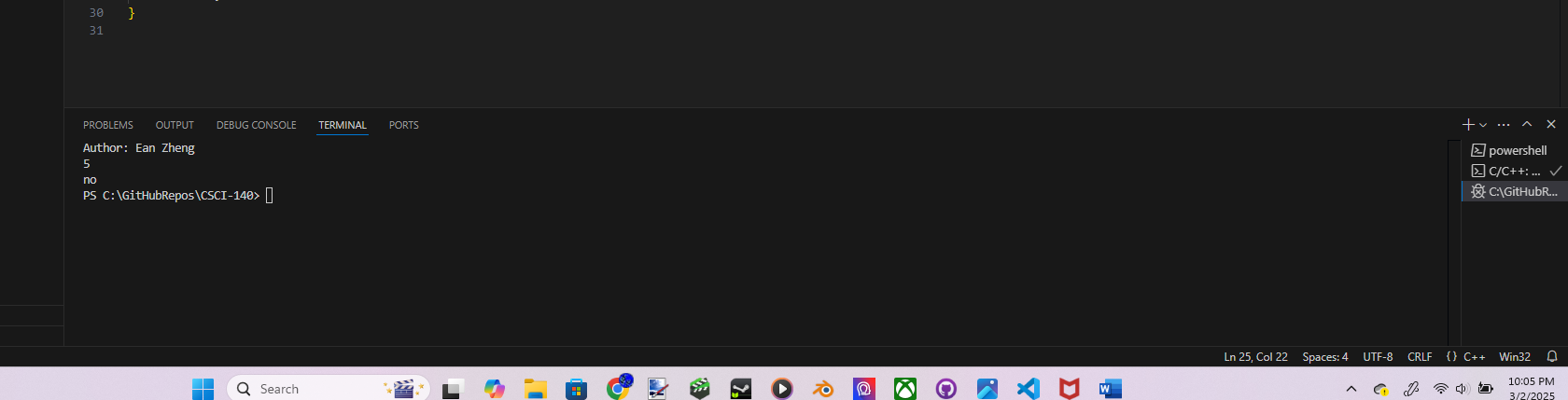
4095

yes



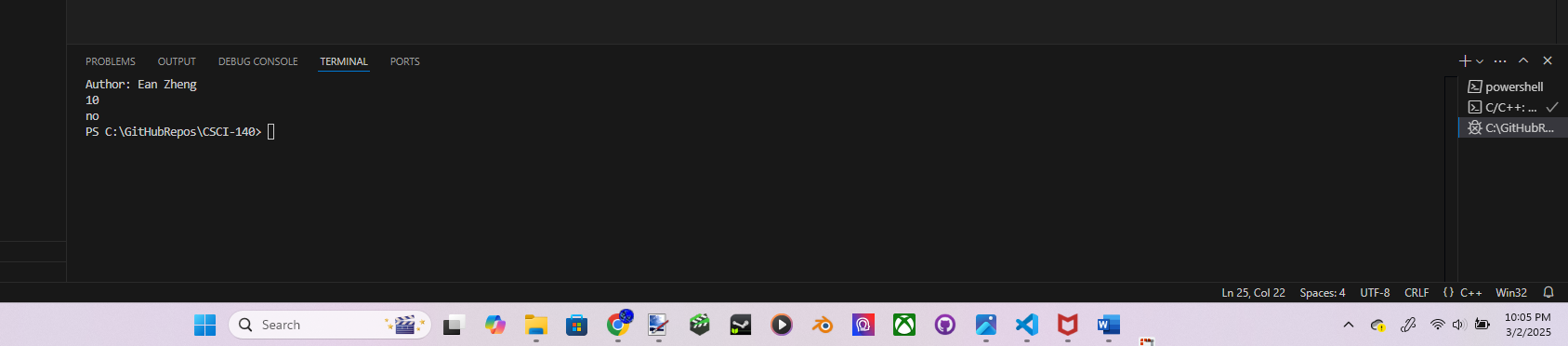
5

no



10

no



5000

no

