import csv

import json

import sys

import xml.etree.ElementTree as ET

# 1. To read 2and 3 columns of a given CSV file and print the content of the column

def quest1():

    print('Read Two columns of a CSV file')

    with open('Data.csv', newline='') as cf:

        data = csv.DictReader(cf, delimiter=',')

        print('Age Salary')

        for r in data:

            print(r['Age'], r['Salary'])

    print('\nRead Three Columns of a CSV file')

    with open('Data.csv', newline='') as cf2:

        data2 = csv.DictReader(cf2, delimiter=',')

        print('Country Age Salary')

        for r2 in data2:

            print(r['Country'], r2['Age'], r2['Salary'])

# 2. To write a Python dictionary to a csv file. After writing the CSV file read the CSV file and

# display the content

def quest2():

    csv\_columns = ['StudentID','Mark 1', 'Mark 2', 'Mark 3', 'Mark 4', 'Mark 5']

    dict\_data = {'Student ID':['1', '2', '3'],

        'Mark 1':[33, 25, 56],

        'Mark 2':[35, 30, 30],

        'Mark 3':[21, 40, 55],

        'Mark 4':[71, 25, 55],

        'Mark 5':[10, 10, 40], }

    try:

        with open("temp.csv", 'w') as csvfile:

            writer = csv.DictWriter(csvfile, fieldnames=csv\_columns)

            writer.writeheader()

            for data in dict\_data:

                writer.writerow(dict\_data)

    except IOError:

        print("I/O error")

    data = csv.DictReader(open("./temp.csv"))

    print("CSV file as a dictionary:\n")

    for row in data:

        print(row)

# 3. To read a given CSV file as a list.

def quest3():

    filename = input("Enter file name: ")

    with open(filename, newline='') as cf3:

        reader = csv.reader(cf3)

        csv\_list = list(reader)

    print("CSV file as a list:\n", csv\_list)

# 4. Display the elements in XML using python with for loop.

def quest4():

    tree = ET.parse('test.xml')

    root = tree.getroot()

    for book in root:

        print(book.attrib)

# 5. With the findall() method work the Parse XML files in Python

def quest5():

    tree = ET.parse('test.xml')

    root = tree.getroot()

    for book in root.findall('book'):

        title = book.find("title").text

        price = book.find("price").text

        print(title, price)

# 6. Write python code to operate on JSON files for the following:

def quest6():

    old\_data = {

        "person" : [

            {

                "firstName": "John",

                "lastName": "Constantine",

                "hobbies": ["magic", "dark arts", "con-artist"],

                "age": 35,

                "children": [

                    {

                        "firstName": "Alice",

                        "age": 6

                    },

                    {

                        "firstName": "Bob",

                        "age": 8

                    }

                ]

            }

        ]

    }

    print("\

    Options\

    1 - loads()\

    2 - load()\

    3 - dumps()\

    4 - dump()\

    5 - Write to Json file\

    0 - Exit\

    ")

    option = input("Enter Number: ")

    match int(option):

        case 0:

            sys.exit(0)

        case 1:

            # a. json.loads()

            json\_string\_loads = json.dumps(data, indent=4)

            print(json\_string\_loads)

        case 2:

            # b. json.load()

            f = open("Data.json", "r")

            data = json.load(f, indent=4)

            for i in data['person']:

                print(i)

            f.close()

        case 3:

            # c. json.dumps()

            json\_string = json.dumps(data, indent=4)

            print(json\_string)

        case 4:

            # dump

            f = open("Data.json", "w")

            json.dump(old\_data, f, indent=4)

            f.close()

        case 5:

            # d. Write JSON to a file

            new\_data = {

                "firstName": "Ezra",

                "lastName": "Muir",

                "hobbies": ["reading", "introspecting", "music"],

                "age": 25,

                "children": None

            }

            with open("Data.json", "r+") as append\_file:

                data = json.load(append\_file)

                data["person"].append(new\_data)

                append\_file.seek(0)

                json.dump(data, append\_file, indent=4)

def main():

    # quest1()

    # quest2()

    # quest3()

    # quest4()

    # quest5()

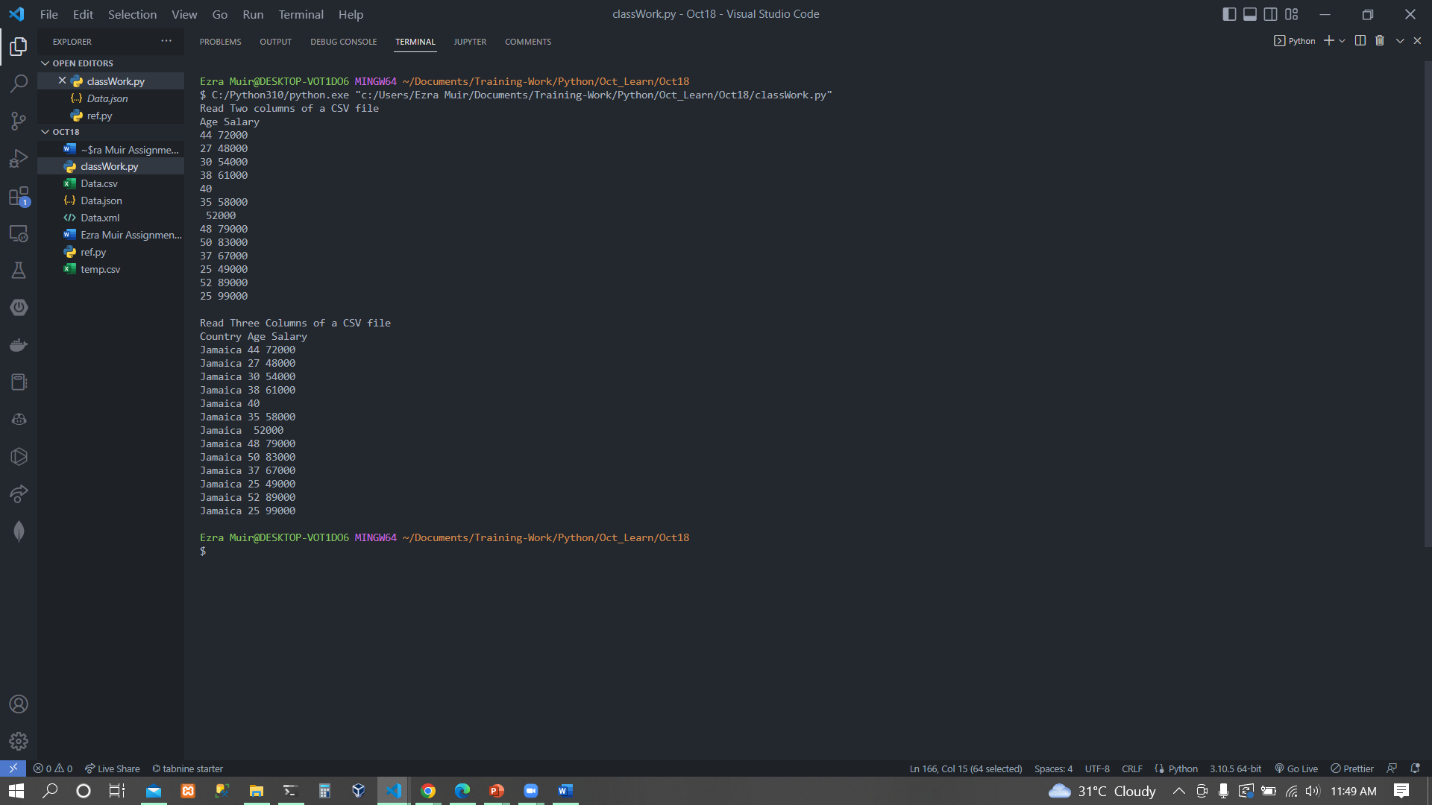
    quest6()

if \_\_name\_\_ == "\_\_main\_\_":

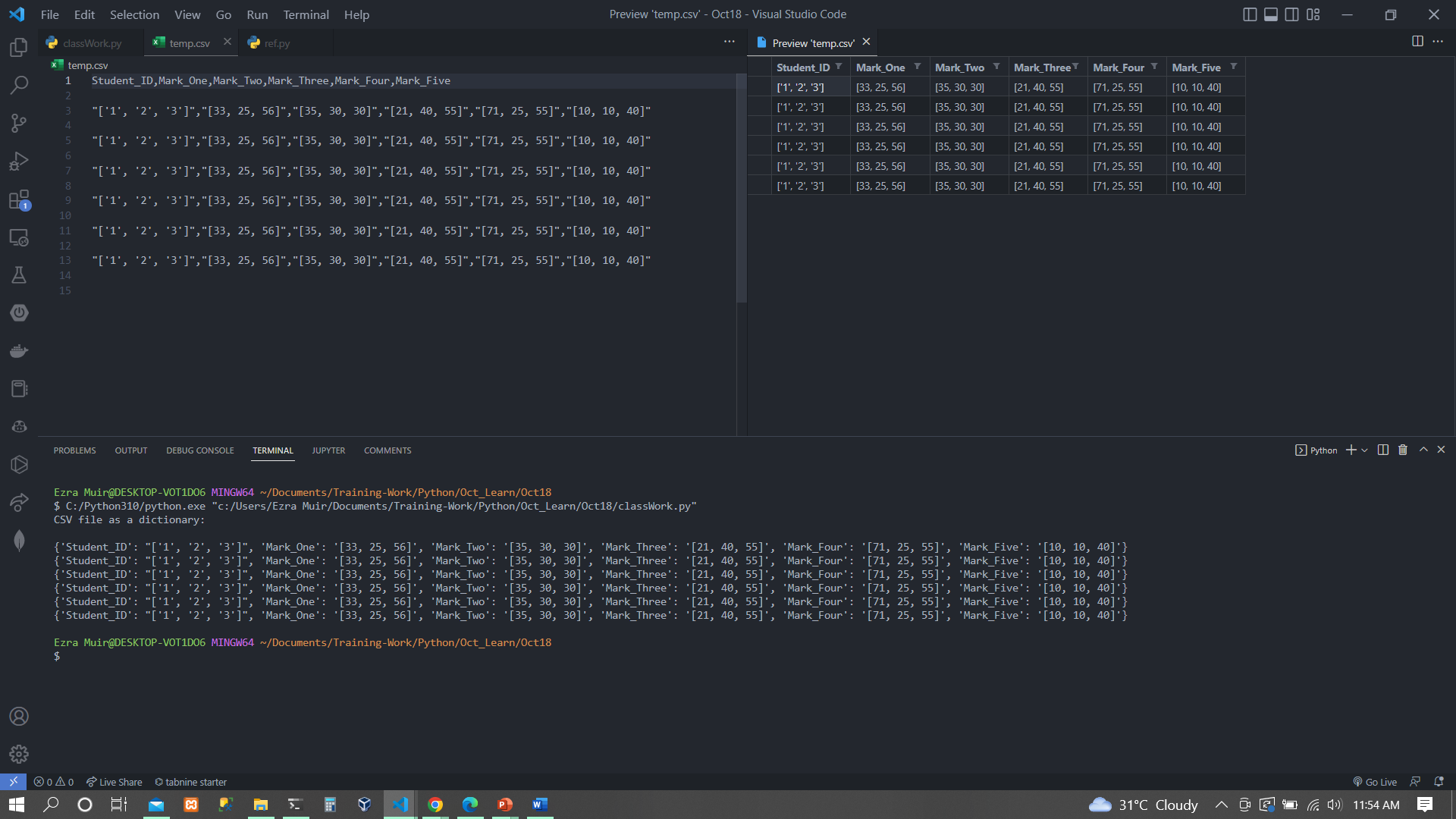
    main()

**OUTPUTS**

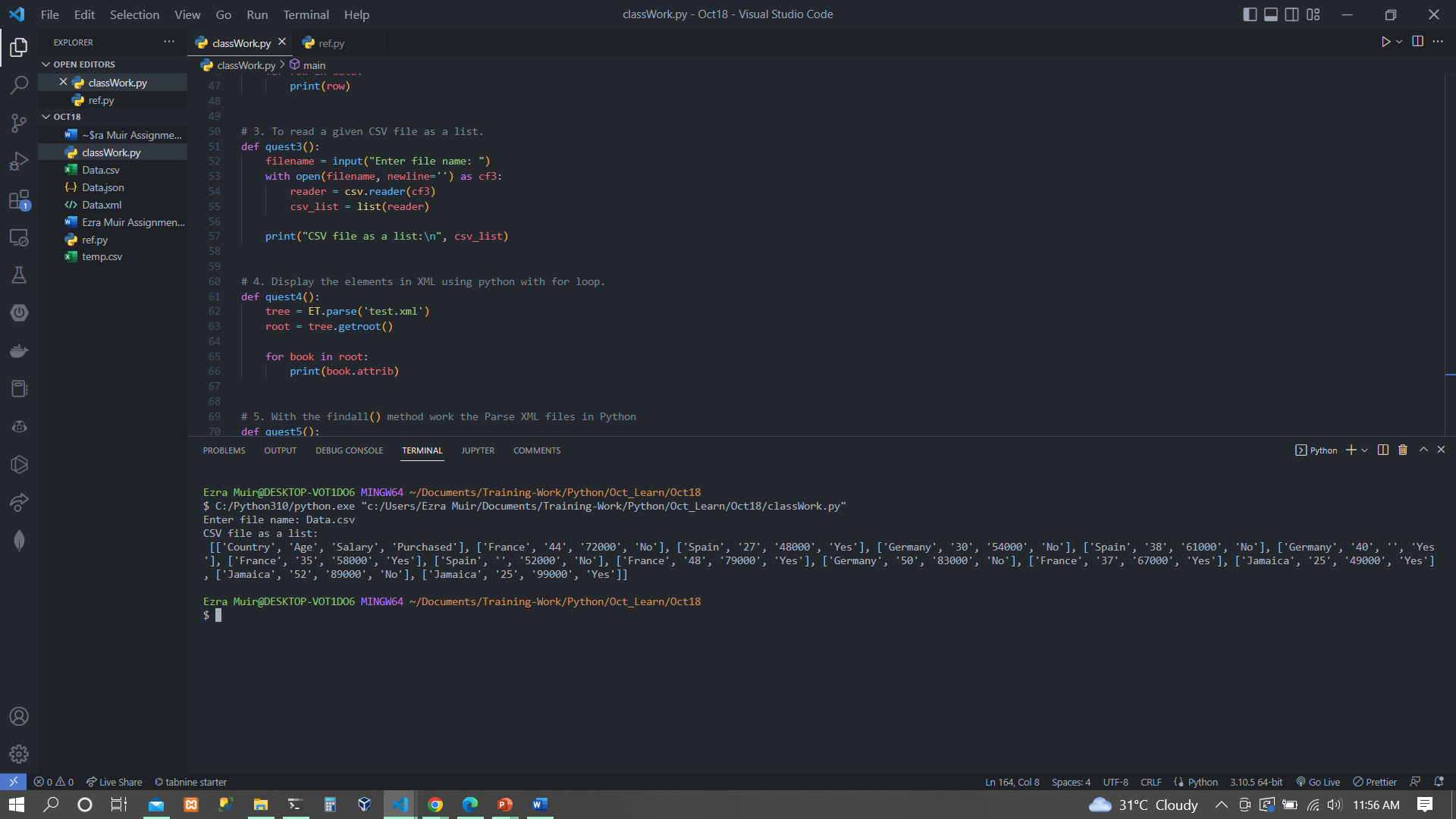
Question 1



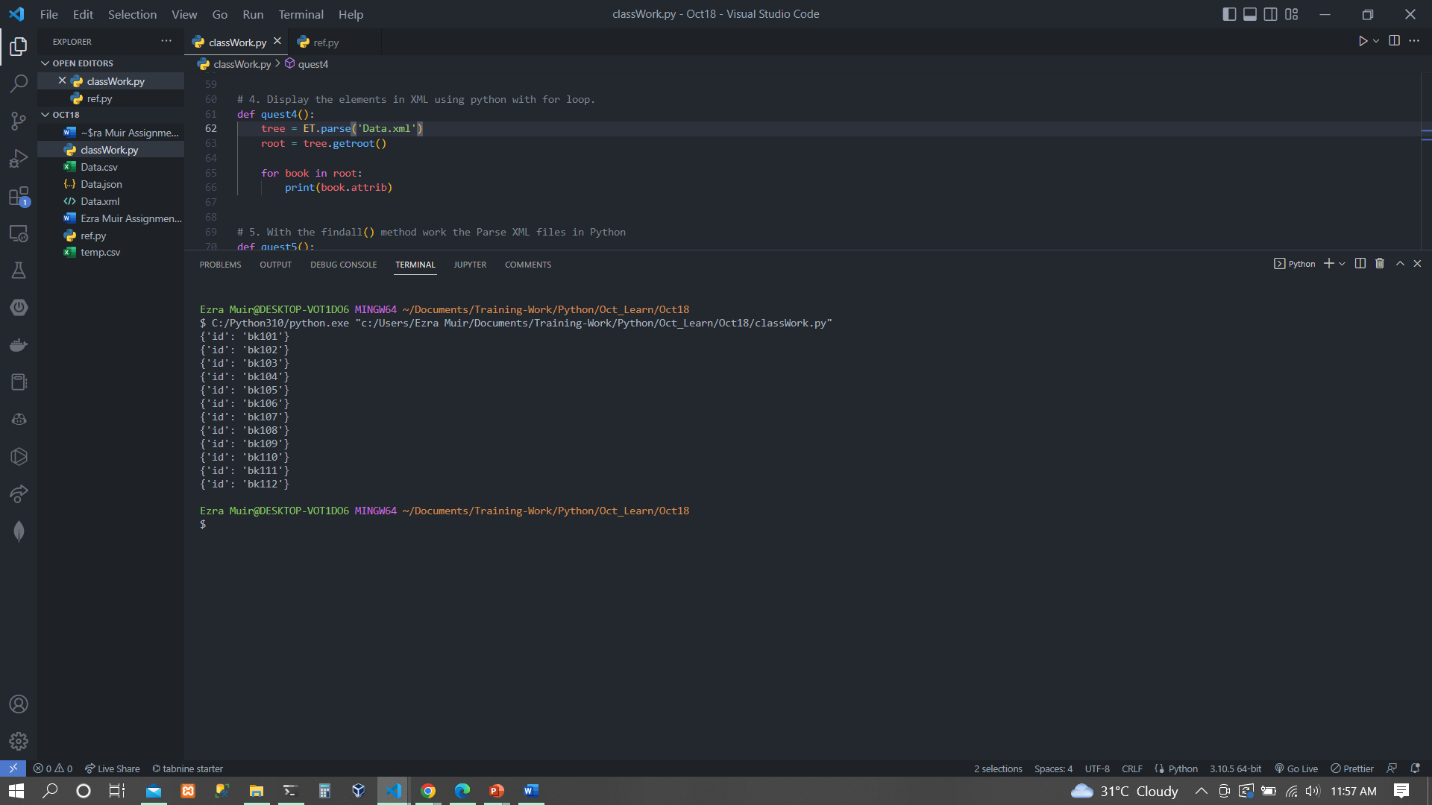
Question 2



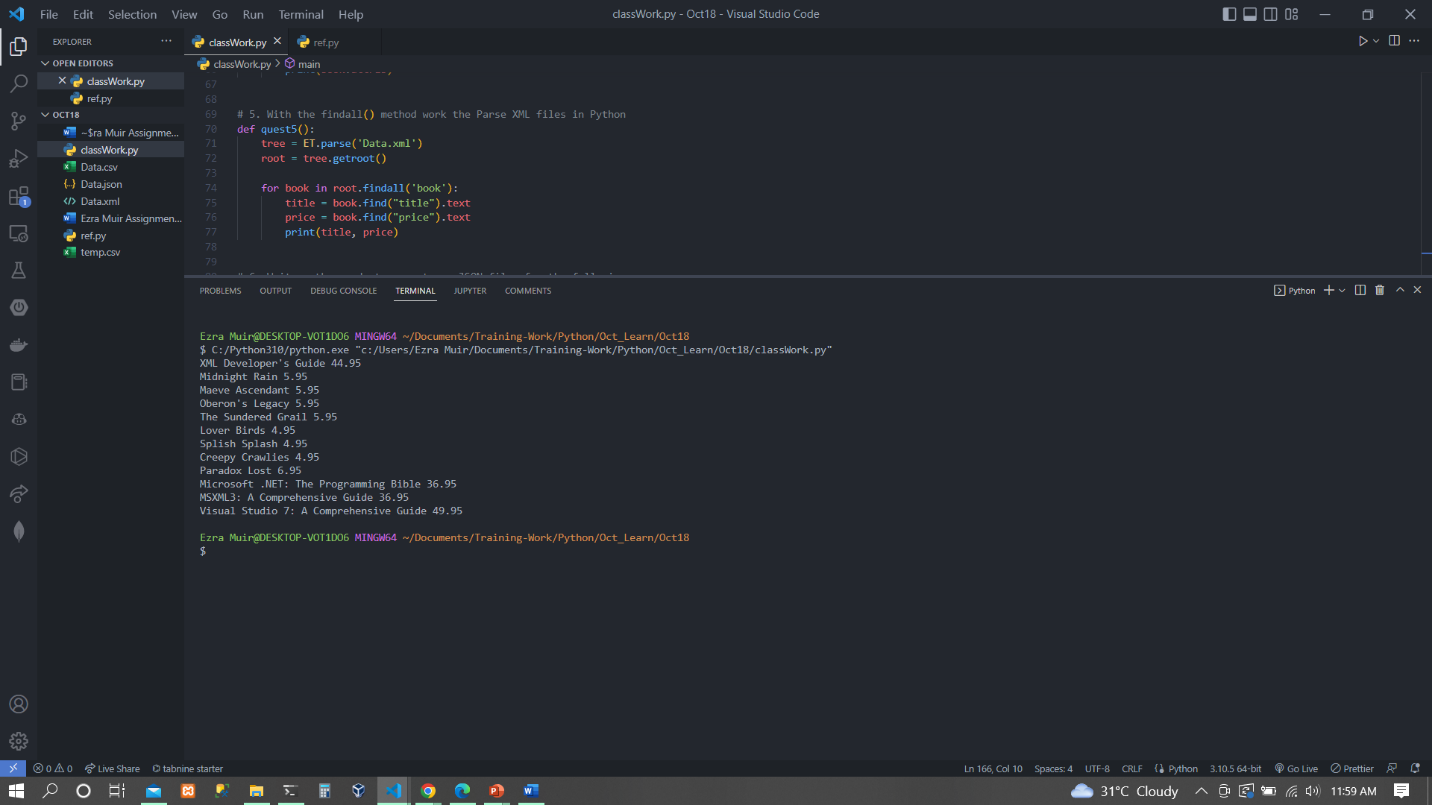
Question 3



Question 4



Question 5



Question 6

