Objectives:

* Writing and reading files
* Try and catch statement
* Intro to GUI Interfaces

**There are 8 challenge exercises with 8 print screens, each worth 12.5%**

Please submit this document for grading when completed… Please work in-groups.

**Project #1** (Writing and Reading files)

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

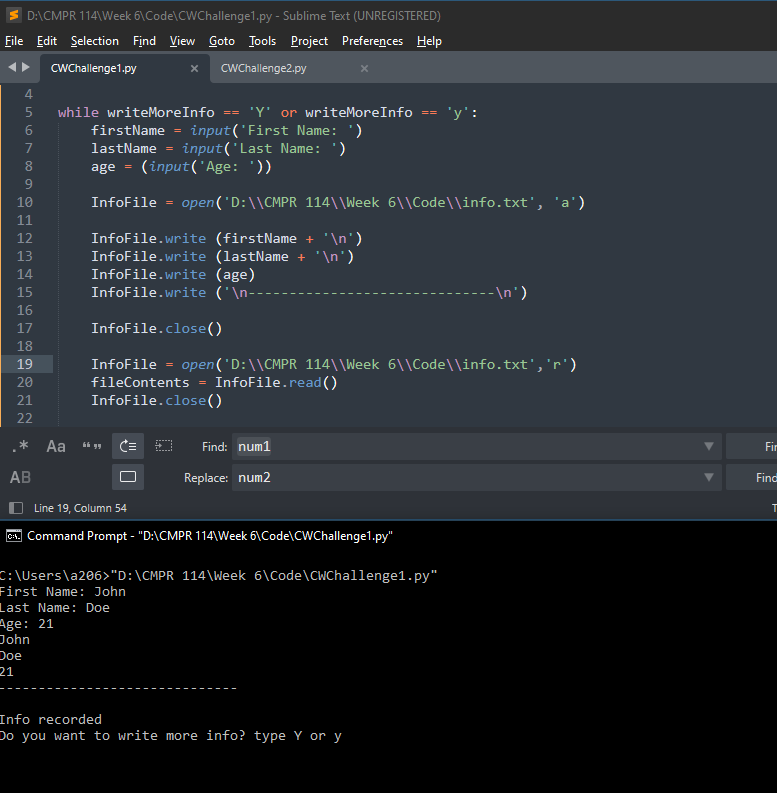
Text

Description automatically generatedA picture containing logo

Description automatically generated

**Challenge Exercise #1:** write a program that will ask the user to enter the first, last name with the age, and write the contents to a file, and then read from the file.

**#1 print screen the running application with code below here.**



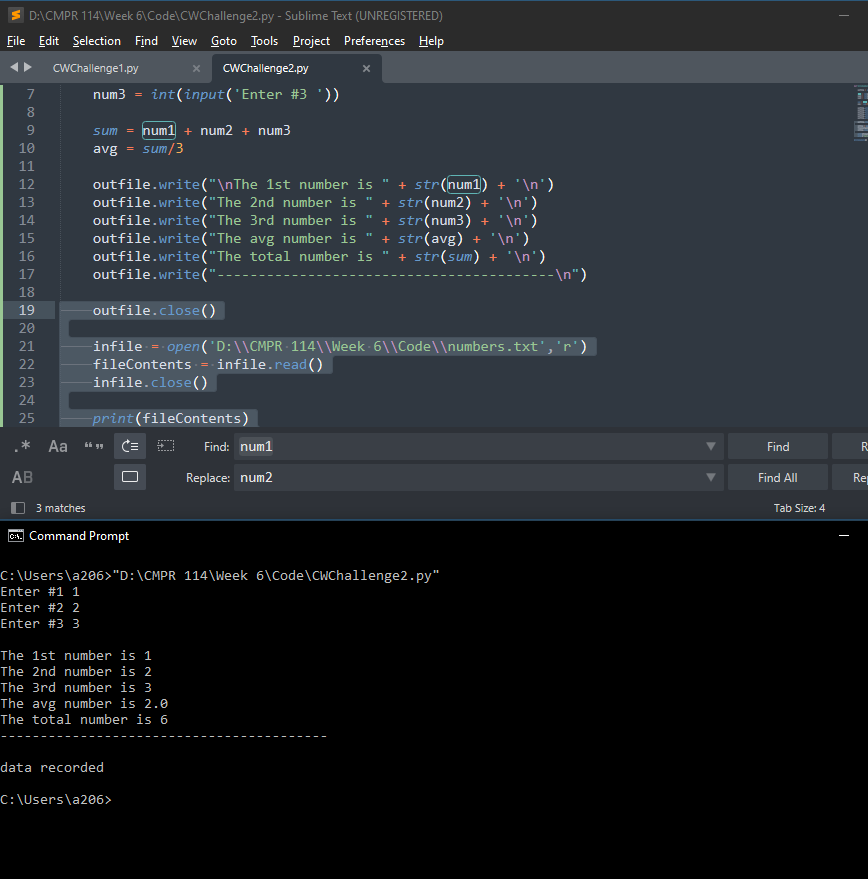
**Project #2** (Writing numeric data)

Text

Description automatically generated with medium confidence

**Challenge Exercise #2:** continuing with project #2, read the data using the print statement.

**#2 print screen the running application with code below here.**



**Project #3** (Writing and reading the Sales Data)

Text, letter

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

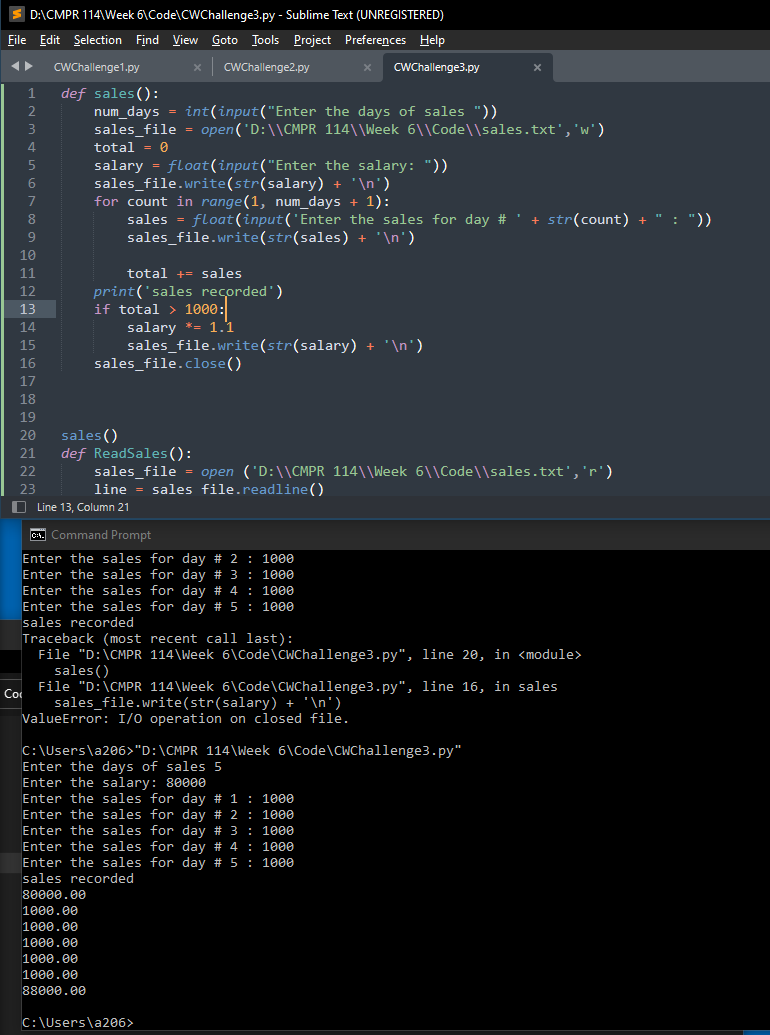
**Challenge Exercise #3:** continuing from project #3, write a program that will ask the user to enter the sales with salary, and if the total sales are greater than 1000 add 10% commission to the salary write and read the data using the print statement.

*Please enter the following test data: enter 80,000 for the salary, enter the days of sale as 5, each sale entered will be 1000, see below*

Text

Description automatically generated

**#3 print screen the running application with code below here.**



**Project #4** (Write Records)

Text

Description automatically generated

**Challenge Exercise #4:** Continuing with project #4, read the content of the employees.txt file

**#4 print screen the running application with code below here.**

**Project #5:** using the try and catch statement with the while loop

Graphical user interface, text, application, email

Description automatically generated

**Project #5 (**Introducing GUI interfaces using Tkinter)

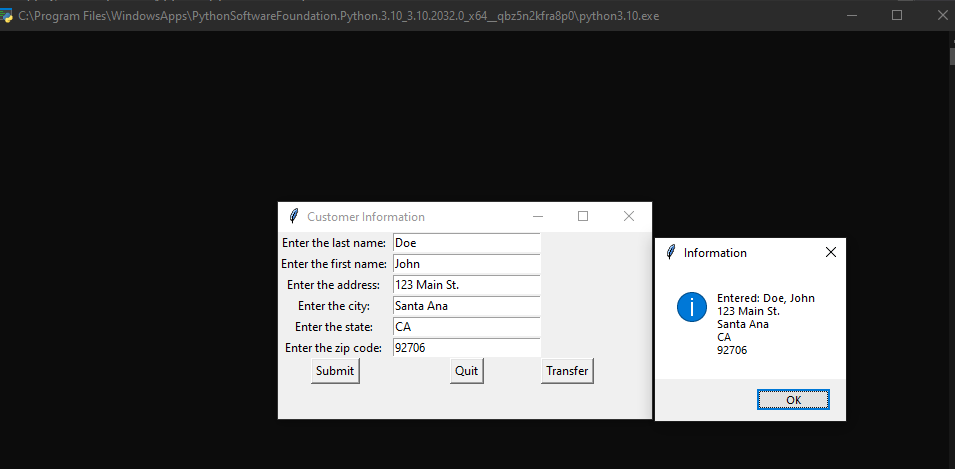
**Text

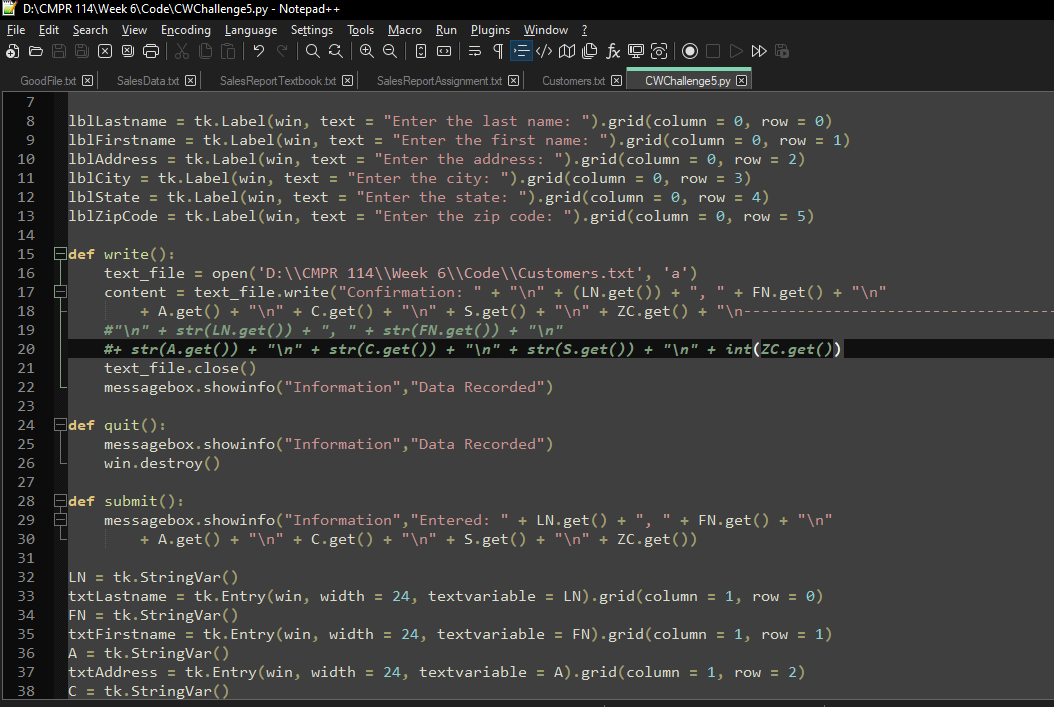
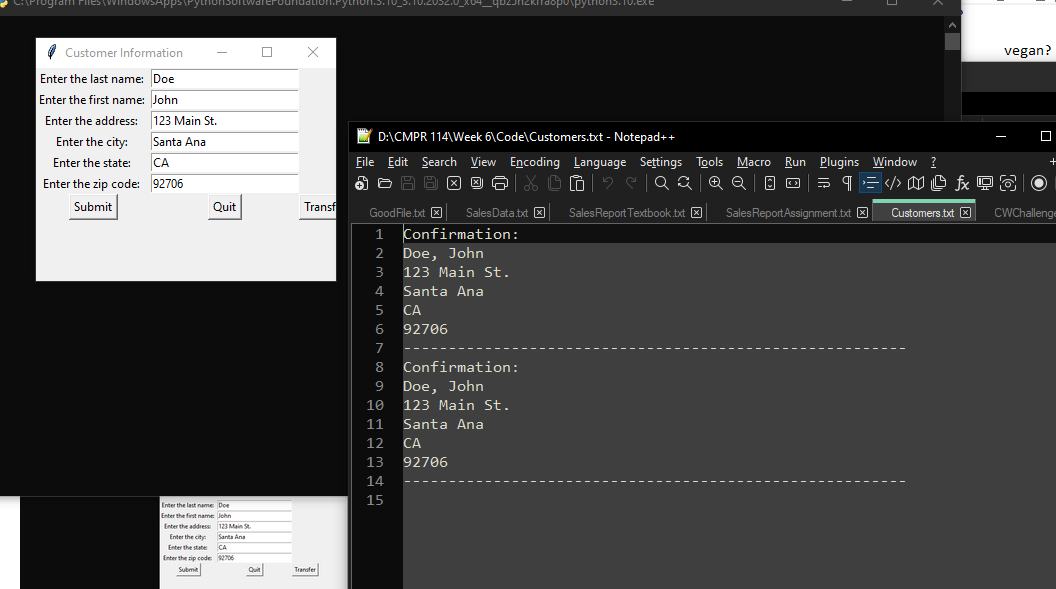
Description automatically generatedText

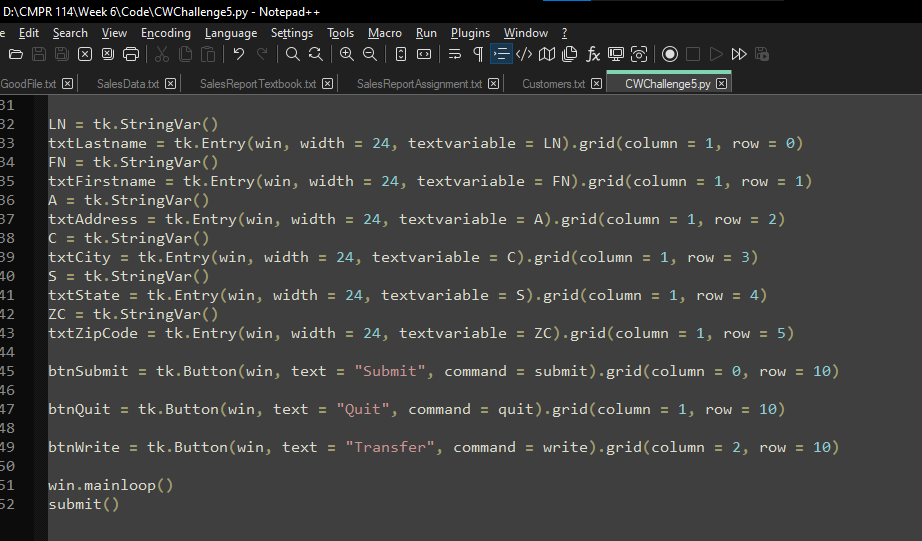
Description automatically generated**

**Challenge Exercise #5:** Continuing with project #5, add the address, city, state with zip code and transfer the information to a text file.

**#5 print screen the running application with code below here.**







**Challenge Exercise #6:** create a **GUI** interface that will write 3 numbers and sum + averages the total into a text file.

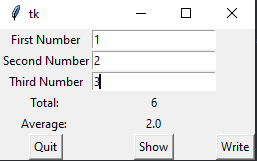
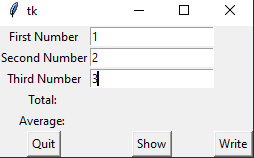
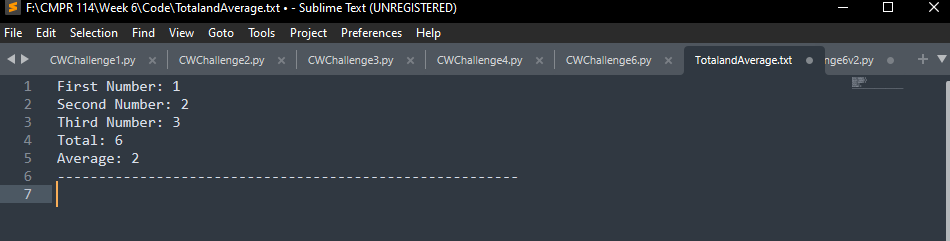
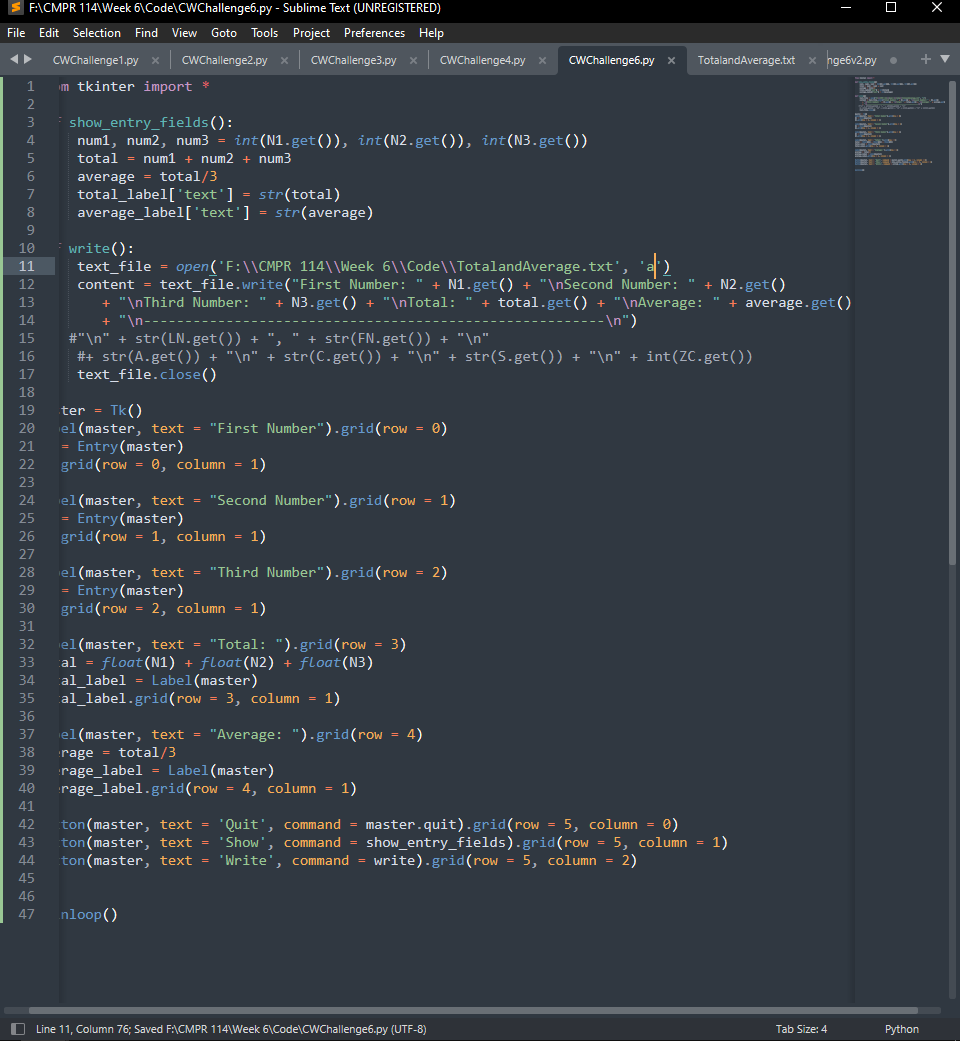
**Example output in the text file**

The three numbers are: 1, 2 and 3

The total is 6

The average is 2

**#6 print screen the running application with code below here.**

from tkinter import \*

def show\_entry\_fields():

num1, num2, num3 = int(N1.get()), int(N2.get()), int(N3.get())

total = num1 + num2 + num3

average = total/3

total\_label['text'] = str(total)

average\_label['text'] = str(average)

def write():

text\_file = open('F:\\CMPR 114\\Week 6\\Code\\TotalandAverage.txt', 'a')

content = text\_file.write("First Number: " + N1.get() + "\nSecond Number: " + N2.get()

+ "\nThird Number: " + N3.get() + "\nTotal: " + "\nAverage: "

+ "\n--------------------------------------------------------\n")

text\_file.close()

master = Tk()

Label(master, text = "First Number").grid(row = 0)

N1 = Entry(master)

N1.grid(row = 0, column = 1)

Label(master, text = "Second Number").grid(row = 1)

N2 = Entry(master)

N2.grid(row = 1, column = 1)

Label(master, text = "Third Number").grid(row = 2)

N3 = Entry(master)

N3.grid(row = 2, column = 1)

Label(master, text = "Total: ").grid(row = 3)

#total = float(N1) + float(N2) + float(N3)

total\_label = Label(master)

total\_label.grid(row = 3, column = 1)

Label(master, text = "Average: ").grid(row = 4)

#average = total/3

average\_label = Label(master)

average\_label.grid(row = 4, column = 1)

Button(master, text = 'Quit', command = master.quit).grid(row = 5, column = 0)

Button(master, text = 'Show', command = show\_entry\_fields).grid(row = 5, column = 1)

Button(master, text = 'Write', command = write).grid(row = 5, column = 2)

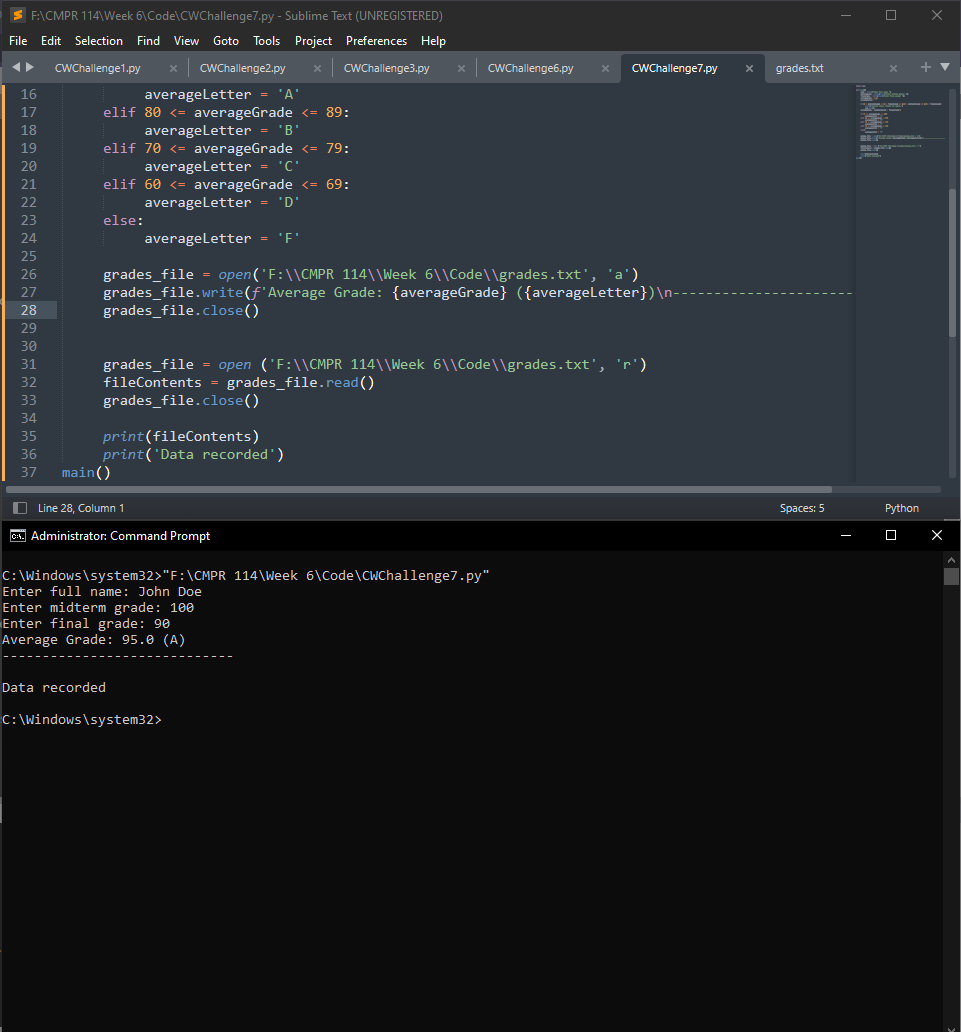
mainloop()

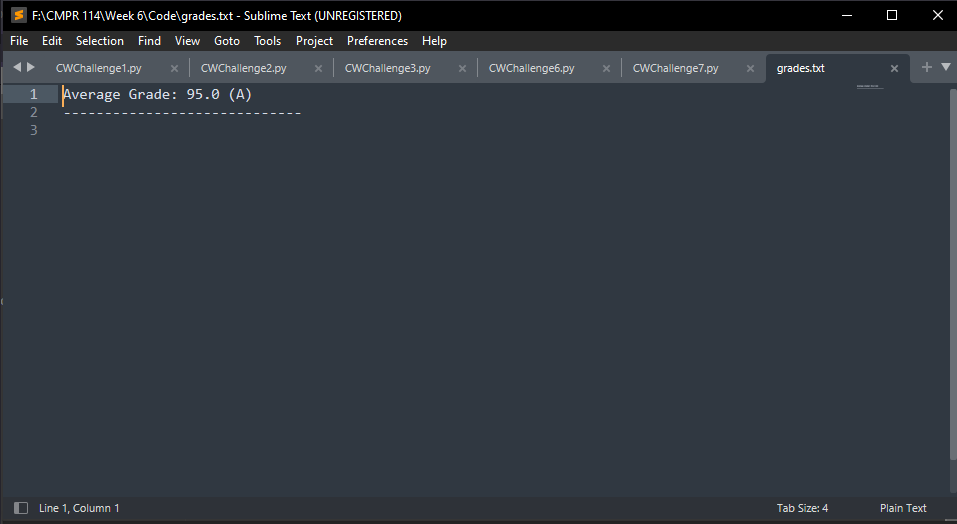
**Challenge Exercise #7:** create a console (**NO GUI**) application that will ask the user to enter the full name, with the grade of the midterm and final exam and **write** the **average** grade with **letter** grade into a text file. Use the table as a guide. Use also the try statement for this application. Make sure to append the data and **read** the content of the file also.

Use the

|  |  |
| --- | --- |
| 90-100 | Letter grade A |
| 80-89 | Letter grade B |
| 70-79 | Letter grade C |
| 60-69 | Letter grade D |
| Under 60 | Letter grade F |

**#7 print screen the running application with code below here.**



import sys

def main():

name = input("Enter full name: ")

midtermGrade = float(input("Enter midterm grade: "))

finalGrade = float(input("Enter final grade: "))

averageGrade = 0.0;

averageLetter = ''

if (0 > midtermGrade) or (0 > finalGrade) or (100 < midtermGrade) or (100 < finalGrade):

print("Invalid input. Please try again.")

sys.exit()

averageGrade = (midtermGrade + finalGrade)/2

if 90 <= averageGrade <= 100:

averageLetter = 'A'

elif 80 <= averageGrade <= 89:

averageLetter = 'B'

elif 70 <= averageGrade <= 79:

averageLetter = 'C'

elif 60 <= averageGrade <= 69:

averageLetter = 'D'

else:

averageLetter = 'F'

grades\_file = open('F:\\CMPR 114\\Week 6\\Code\\grades.txt', 'a')

grades\_file.write(f'Average Grade: {averageGrade} ({averageLetter})\n-----------------------------\n')

grades\_file.close()

grades\_file = open ('F:\\CMPR 114\\Week 6\\Code\\grades.txt', 'r')

fileContents = grades\_file.read()

grades\_file.close()

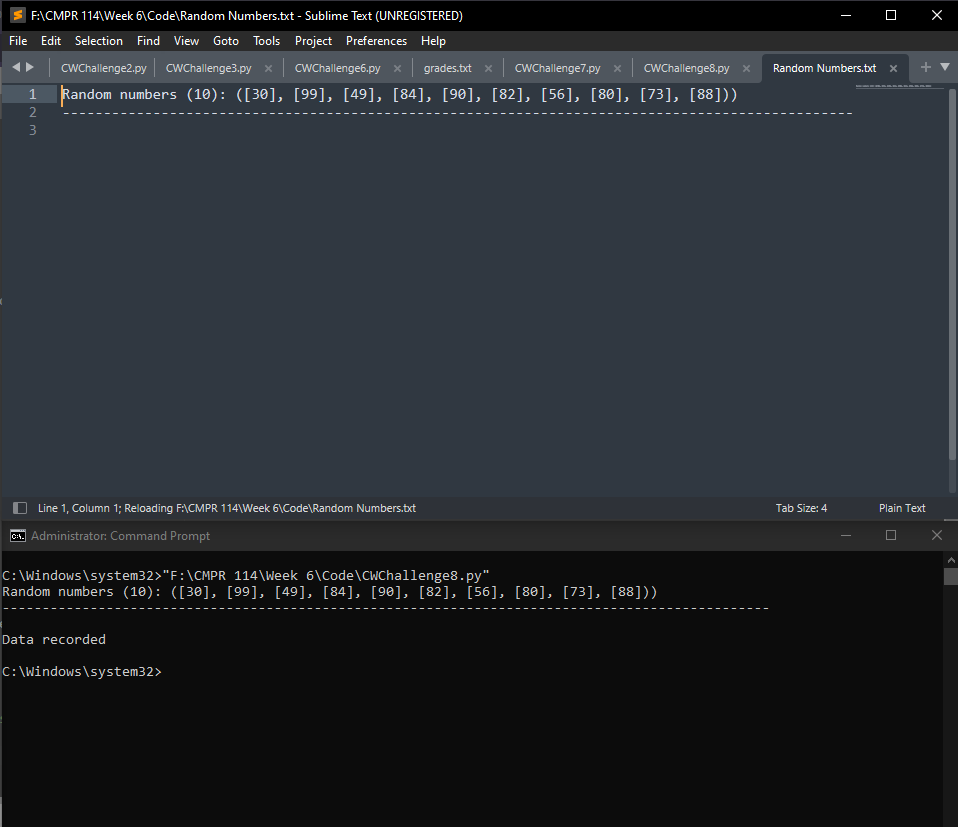
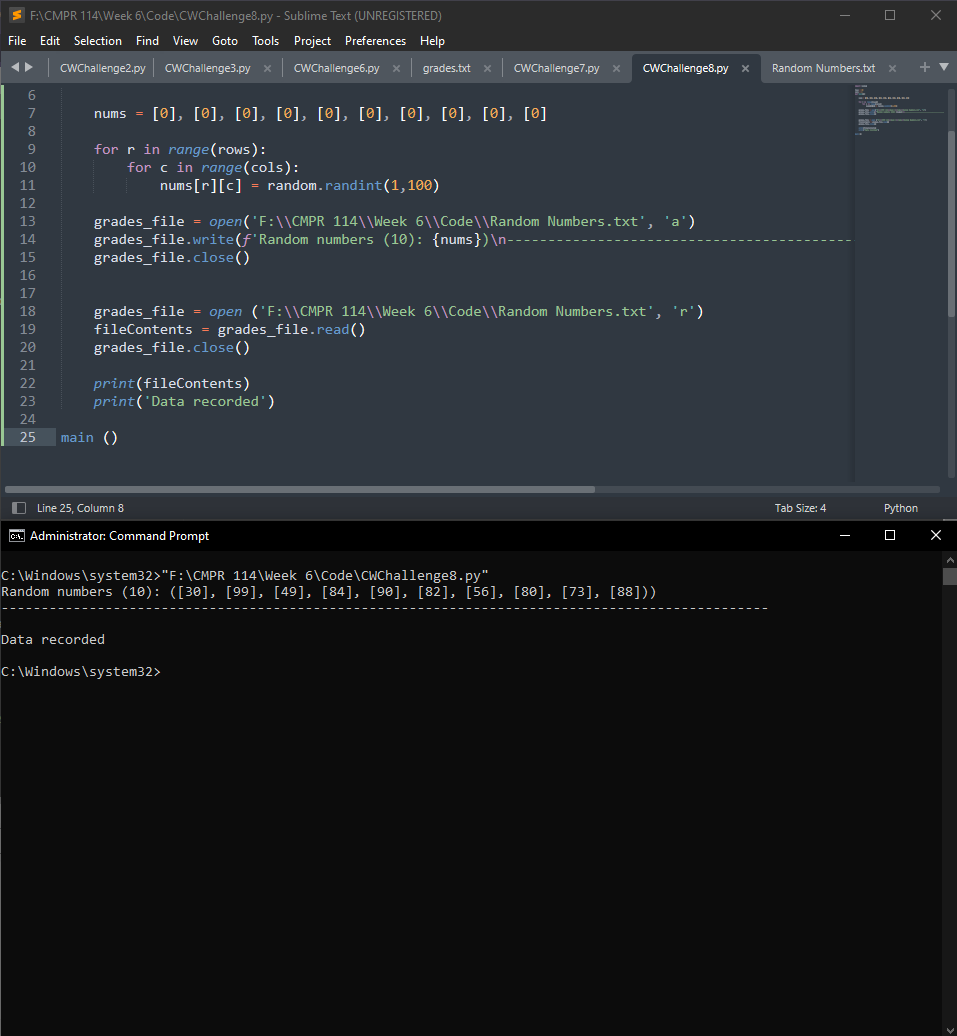
print(fileContents)

print('Data recorded')

main()

**Challenge Exercise #8:** create a console application (**NO GUI**) that will write random numbers from 1-10 into a text file and read the contents.

**#8 print screen the running application with code below here.**

import random

rows = 10

cols = 1

def main():

nums = [0], [0], [0], [0], [0], [0], [0], [0], [0], [0]

for r in range(rows):

for c in range(cols):

nums[r][c] = random.randint(1,100)

grades\_file = open('F:\\CMPR 114\\Week 6\\Code\\Random Numbers.txt', 'a')

grades\_file.write(f'Random numbers (10): {nums})\n------------------------------------------------------------------------------------------------\n')

grades\_file.close()

grades\_file = open ('F:\\CMPR 114\\Week 6\\Code\\Random Numbers.txt', 'r')

fileContents = grades\_file.read()

grades\_file.close()

print(fileContents)

print('Data recorded')

main ()

**Submit this document to Module 6 Class Exercise.**