Name Kweku Andoh Yamoah ID:71712022 Final Project.

Problem: My program is to help a hospital save an appointment for patients who have requested for appointments. I have a doctor that can handle multiple patients and where I set up a scheduling program where a doctor can only handle 8 patients during an 8-hour work day after noon.

How Program Works.

My program is divided into four parts. The four parts were constructed using object-oriented design how the part works are as follows.

Part I(class Patient): In this first part the goal is to take the first name of a patient and save it for future appointment booking. There are two getters getFName and getLName which returns the first name and the last name of the patient respectively. Under this object, I created a helper function called patient_file. Here, I open a text file called "patientApiontment.txt" for reading. I then loop through the lines of the file. I open the file "patientApiontment.txt" for appending again where I write the new instance of the patient class to the file. I have set Boolean conditions which control the logic behind the writing. If found ="False" I print to the console that the patient has been saved successfully.

Part II(class Interface):

The goal of class interface is to control what a user can do with the program. Class Interface takes an instance of class Patient as a parameter. I have six methods here:

a. def appointment(): this is a function that I created to book appointments for patients in the "patientApointment.txt" file. The text file is opened for reading and splitting of contents. I assign the results to a variable called lines where I find a random choice of lines. I imported random from the onset of the program. I again assign this result to a variable called mypatient where a split and assign the results to variables called f_name and l_name respectively. I then call on a function called doc_chosen where I get a doctor, a day and time. I then print these five results to a new file called "Appointment.txt" which I had opened for appending. I return the f_name,l_name,doc,day,time variables to be used in future.
b. def doc_day(): This function is to assign a doctor to a day read from a file with the data. I first display the doctors available for assignment. I then ask for two inputs from the user one is the name of the doctor in the same format that has been displayed and the second one is the name of a weekday fully. I then open the "Doctor file.txt" for reading I loop through the lines where I split at a ",[". I then evaluate the second variable l_days which ideally should hold a string, but due to how the data is arranged I make python do this for me. I then check if the inputted doctor name is in my file, if true, I again check if the inputted is in my list. If this condition is true, I then return the variables doc and check_day which hold the doctor name and dya selected respectively.
c. def display_slots(): This function display static time slots for the day to see. And I call it at any point of the Interface object where it is needed.
d. def doc_chosen(): This is a method to save the doctor, the day and the time to a file called "Doc_chosen.txt". I first call the doc_day function and assign it to two variables called doc and day. I call on the display slots function to show the static slots to the user in the console. I open the "Doc_chosen.txt" for reading. I provided. have a list of strings called slot which hold the time times that can be booked. I take an integer input from the user which specifies the time for an appointment. I then read the lines of the text file and close it. I open the same file for appending. And I assign a Boolean value to a variable called found. I loop through the lines read mypatient, and I assign the to three holding variables after splitting. I then check if the doc, day and the user optin is in my file. If it is True, I print a notice to the console for the user to see, I set found to True now, and I break out of the condition. If found remains False, I print the results to the text file opened and I return them.
e. def dis_docbookings(): I created this method to this paly the doctors booked, the day and the time is chosen. Here, I open the "Doc_chosen.txt" file where I loop through the lines. I split the results and hold them in three variables respectively. I now print these variables which have formatted to make the user understand it easily.
f. def del_appointment(): This is the last method which allows the deletion of an appointment. I call the display slots function. I ask the user for five inputs, the first name, the last name, that doctor name, the appointment day and the time booked which I hold in five variables. I then open the "Appointment.txt" file where I split the results. I now open the file for writing, if the user inputs are not the same as my file results I write it back to the file.

Part III(class App): The goal of this class app is to handle the use of the program. I accept the class Interface as a parameter. I then print a menu of what the user can do with my program. I have a helper function which controls what the user can do based on the input provided .

def main(): This is a function I created to create instances for objects I have created and the methods they hold.

Limitations:

- The program will not run if the user fails to input the same input as I have aligned in the implementation of the code.
- The program lacks a graphical user interface.