CS 115 - Introduction to Programming in Python

Lab Guide 08

Lab Objectives: Inheritance

Notes:

a) Upload your solutions as **a single .zip file** to the Lab08 assignment for your section on Moodle **by 17:30 on Friday, December 11**. You must use the following naming convention: Lab08_Surname_FirstName.zip where Surname is your family name and FirstName is your first name.
b) Solutions sent through email will not be accepted.
c) You should only use functionality covered in CS115 in your solution.
d) Include a docstring for your functions.

1. Create a class, **`Patient`**, with the following data attributes and methods. Note all data attributes and class variables should be private.

Data Attributes:

- **pName**: stores the string name of the patient.
- **isInsured**: boolean field that indicates if patient has private insurance.
- **coveragePercent**: stores the percent (as decimal value) of the patient's insurance coverage. Zero if not insured.

Class Variable:

- **hospitalFee**: stores the fee for the hospital visit, 200TL.

Methods:

- **__init()__**: initializes the pName, isInsured, coveragePercent (default parameter set to zero if not passed) to values passed as parameters. Should initialize coveragePercent using the set method.
- **Get and set methods** for all data attributes. The set method for coveragePercent should only set the variable if the value passed as a parameter is a positive value.
- **Get method to return the value of the __hospitalFee.**
- **__repr()__**: returns a string representation of a Patient object formatted as shown in the sample run (includes patient name and insurance information only).
- **calculateFee ()**: calculates and returns the amount of the hospital fee the patient must pay. If the patient is insured, deduct the insurance portion.

2. Create a subclass, **`Outpatient`**, by extending the superclass **`Patient,`** with the following data attributes and methods. Note all data attributes should be private.

Data Attributes:

- **polyClinic**: stores the string name of the poly clinic for the patient's appointment.
- **doctorName**: stores the string name of the doctor the patient will visit.
- **appointmentDate**: stores the date of the appointment.
- **appointmentTime**: stores the time of the appointment.

Methods:

- **`__init()__:`**
  - Takes the following parameters: name, insurance, appointment date, appointment time, poly clinic, doctor name, and coverage percent (default 0.0) as parameters.
  - Initialize the Patient data using the super class `__init__` method. If the polyclinic is Dentistry or Optometry, the coverage percent passed as a parameter should be divided by 2.
  - Initialize appointment date, time, doctor and poly clinic to the parameter values.
  - Use the set method to initialize the appointment date.

- **Get and set methods for Outpatient attributes**:
  - **`setAppointmentDate()`** takes a string as a parameter (assume 'YYYYmmdd') and converts it to a date object using the datetime module.
  - **Example:**
    datetime.datetime.strptime(varName, '%Y%m%d').date() converts the given varName string to a date, where %Y indicates the position of the 4 digit year, %m the two digit month, and %d the two digit date.

```
today = datetime.datetime.strptime('20200203', '%Y%m%d').date()
today
Out[13]: datetime.date(2020, 12, 03)
```

- **__lt()__**: compares two Outpatients by their appointment date and time. If self has an appointment date and time before other, return True, else return false.

- **__repr()__**: returns a string representation of an Outpatient object. The method should call the Patient __repr__ to get the Patient data, and append the Outpatient data, formatted as shown in the sample run.

3) Write a script **`PatientApp`** with the following functions:
   - **schedulePatients():** takes a string filename as parameter and returns a list of Outpatients containing the patient data from the input file.
   - **The script** should do the following:
     - Schedule the patients in the file `patients.txt` using the function above.
     - Sort the list of patients according to their appointment times.
     - Display the list of patients.

Sample Run:

```
[
Appointment Date: 2020-12-08 15:30
Patient Name: Hale Sert Insurance: (yes)
Poly Clinic: Neurology (Dr. Melis Koç)
Fee: 20.0
,
Appointment Date: 2020-12-10 10:15
Patient Name: Ece Top Insurance: (yes)
Poly Clinic: Dentistry (Dr. Ali Ayhan)
Fee: 130.0
,
Appointment Date: 2020-12-11 08:30
Patient Name: Su Kara Insurance: (no)
Poly Clinic: Dematology (Dr. Irem Basar)
Fee: 200
,
Appointment Date: 2020-12-11 11:00
Patient Name: Lale Kaleci Insurance: (yes)
Poly Clinic: Cardiology (Dr. Ayla Güner)
Fee: 120.0
,
Appointment Date: 2020-12-17 10:30
Patient Name: Oya Ak Insurance: (yes)
Poly Clinic: Cardiology (Dr. Veysel Karakuş)
Fee: 40.0
,
Appointment Date: 2020-12-17 15:30
Patient Name: Mahmut Pembe Insurance: (yes)
Poly Clinic: Optometry (Dr. Elif Som)
Fee: 110.0
,
Appointment Date: 2021-01-06 12:15
Patient Name: Ali Uzun Insurance: (no)
Poly Clinic: Neurology (Dr. Jale Tunç)
Fee: 200
,
Appointment Date: 2021-02-06 15:30
Patient Name: Emel Kaya Insurance: (no)
Poly Clinic: Optometry (Dr. Mehmet Keskin)
Fee: 200
]
```