CMSC 201 Section 60
Homework 6
Fall 2024

Due Date:

No later than midnight Eastern time on Monday, December 9, 2024

Value:

This assignment is worth 40 points. Each of the two parts is worth 20 points.

Assignment:

General:

The assignment must be done in accordance with the rules for all assignments; e.g. comment blocks with names, dates, etc.

Code:

For both parts, you can use the code on the following page to get started. This code is available on GitHub as the file ***hw6.py***

```
"""
Starter code for CMSC 201 Section 60 Homework 6. This code begins the
definition of the class "Patient"
and its subclass, "Pediatric_Patient".  An init method and one other method
are defined for
each class.
"""


class Patient:
   def __init__(self, name, age, gender, insurance,record):
       self.name = name
       self.age = age
       self.gender = gender
       self.insurance = insurance
       self.record = record

   def admit_patient (self, adm_date):
       self.record["admit"] = adm_date

# def discharge_patient(self, discharge_date):
# write the body of this method:
# add a field to "record" - call it "discharge" and give it the value
"discharge_date"

class Pediatric_Patient(Patient):
   def __init__(self, name, age, gender, insurance, record, guardian):
       super().__init__(name, age, gender, insurance, record)
       self.guardian = guardian


if __name__ == "__main__":

   #create a patient named "Joe"
   patient_1 = Patient("Joe", 54, "M", {"co":"InsCo", "acct":"12345"}, {})
   # now add an admission date
   patient_1.admit_patient("02/15/2024")

   # now check to see if it worked
   print(patient_1.record)

   p2 = Pediatric_Patient("Sue",7,"F",{"co":"InsCo", "acct":"12345"}, {},
"Steve" )
   print(p2.guardian)
```

Part 1:

Write an additional method for class Patient to add a discharge date for the patient. A short skeleton for this method is provided on lines 19 through 21 of the starter code. Turn that commented code into actual method code.

Your method must add an entry into the Patient.record field that includes the discharge date. Do not delete the admission date; add the discharge date to it.

To test: immediately after this line in your main program:

```
patient_1.admit_patient("02/15/2024")
```

add the following line:

```
patient_1.discharge_patient("02/18/2024")
```

and run the program. The statement

```
print(patient_1.record)
```

Should now produce this output:

```
{'admit': '02/15/2024', 'discharge': '02/18/2024'}
```

Part 2:

All pediatric patients must have a designated adult guardian in their records. Creating an instance of subclass "Pediatric_Patient" with an empty string in the guardian field would be an error. Add a method to the definition of Pediatric_Patient to check for this, and force the user to enter a valid name in that field.

Use the starter code. Modify class Pediatric_Patient as follows:

```
class Pediatric_Patient(Patient):
   def __init__(self, name, age, gender, insurance, record, guardian):
       super().__init__(name, age, gender, insurance, record)
         if guardian == "":
              guardian = Pediatric_Patient.error_check(self)
       self.guardian = guardian
```

The two bolded and italicized lines are new.. They check to see if the guardian is an empty string - that is, there is no guardian defined for this child. If that's the case, a call is made to a method, “error_check’ which will return a valid guardian name. Define that method, in the the class Pediatric_Patient. It should start:

```
      def error_check(self):
```

And then contain a loop that executes until the user enters a non-empty string as the guardian. That string is returned.

To test your code, add the following lines at the end of the main program:

```
      p3 = Pediatric_Patient("Maria",9,"F",{"co":"InsCo", "acct":"56789"}, {},
      "" )
      print(p3.name, p3.age, p3.guardian)
```

The program should print the error message and prompt for the name of the guardian. The first time that occurs, hit “Enter” to enter an empty string. This is testing that the loop works.
The second time you are prompted to enter a name, enter “Mark.” When your program ends it should print the following output:

```
      There must be a designated adult guardian for each pediatric patient
      Please enter the name of the adult guardian
      There must be a designated adult guardian for each pediatric patient
      Please enter the name of the adult guardianMark
      Maria 9 Mark
```

The first error message/prompt is caused by the line “p3 =...” which contains an empty string for the guardian. The second error message/prompt is caused by the fact that you just hit “Enter” at the first prompt. The final line is the correct output once you’ve entered the guardian name.