CSC 242: Introduction to Computer Science II

Assignment 2

Due Tuesday, April 14th, 11:59pm

Reading

Read Chapter 8 in **Introduction to Computing using Python: An Application Development Focus** by Ljubomir Perković.

Logistics

Please read the Academic Integrity policy beginning this assignment. It is crucial that you understand what kind of collaboration is allowed and what kind is disallowed on assignments.

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in a comment at the top of the assignment, and the number of collaborators on any assignment **may not exceed two other people**. You must also submit a comment in your submission for each assignment that describes in detail how each collaborator contributed to the assignment. If you did not collaborate with anyone on the assignment, you must include a comment that says that. You may not under any circumstances discuss the assignments with classmates' other than your identified collaborators. Working so closely with anyone other than your identified collaborators Mr. Zoko, or Mr. Feingold, so as to produce identical or near identical code is a violation of the Academic Integrity policy. This policy will be strictly enforced.

Please include the following with your assignment submission:

1. A comment at the top of your Python file identifying any classmates with whom you discussed or in any other way collaborated on the assignment. You may work (directly or indirectly) with **no more than two** other people.
2. Add a comment at the top of your Python file that describes for each person what they contributed to the assignment. This must be at least 2-3 sentences and be **very specific and detailed**.

A submission that does not include a list of collaborators and a comment indicating how you collaborated with classmates will earn a 0. If you worked alone you must put a comment at the top of your file that indicates that or you will also receive a 0. There will be no exceptions to this rule.

Again, you are subject to all the rules specified in the Academic Integrity policy. Please read it carefully before beginning this assignment.

Assignment

Begin the assignment by downloading the template **csc242hw2.py** found on the D2L site. You will complete the classes and function header found there.

Part 1 (50 points) Implement a class **Patient** (that is a subclass of the object class) the description of each of the methods are found in the template file.

The patient class will calculate the BMI of the patient. Information about BMI can be found here:
https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/english_bmi_calculator/bmi_calculator.html

The formula is: BMI = 703 × weight (lbs) / [height (in)] ^ 2
Hint: There are 12 inches in a foot.
Hint 2: You can call another method within the same class by using self.methodname()

The ranges we are using are:

- A BMI less than 18.5 the person is 'Underweight
- Person is greater than or equal 18.5 and less than 25 they are 'Normal Weight'
- Person is greater than or equal to 25 and value less than 30 they are 'Overweight'
- Person BMI is greater than or equal to 30 they are 'Obese'

Temperature ranges are considered:

- Low if the temp is less than 97.7
- Normal if it is between 97.7 and 99.5
- High if it's over 99.5

The following shows how the **Patient** class and its methods could be used:

Default:

```
>>> p=Patient()
>>> p.getBMIDescription()
'Obese'
>>> p.getBMI()
703.0
>>>
```

Another example:

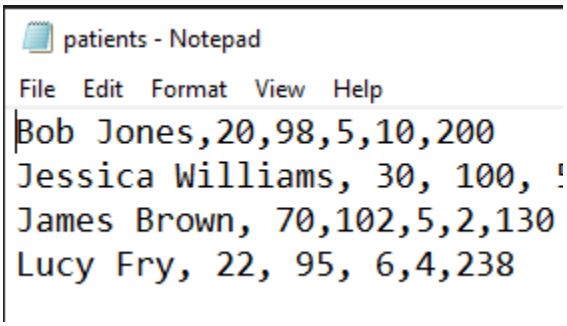
```
>>> p=Patient('Bob',30,99, 6,4,200)
>>> p
Patient('Bob',30,99,6,4,200)
>>> str(p)
'Bob is 30 years old.  The patient is Normal Weight and their temp is Normal.'
>>> p.getBMI()
24.342105263157897
>>> p.getHeight()
(6, 4)
>>> p.getWeight()
200
>>> p.getTemp
<bound method Patient.getTemp of Patient('Bob',30,99,6,4,200)>
>>> p.getTemp()
99
>>> p.getAge()
30
>>> p.getBMIDescription()
'Normal Weight'
>>> p.tempRange()
'Normal'
>>>
```

Another Example:

```
>>> p=Patient()
>>> p.setAge(20)
>>> p.setHeight(5,10)
>>> p.setWeight(140)
>>> p
Patient('',20,98.6,5,10,140)
>>> str(p)
' is 20 years old.  The patient is Normal Weight and their temp is Normal.'
>>> p.getBMIDescription()
'Normal Weight'
>>> p.getBMI()
20.085714285714285
>>> p.tempRange()
'Normal'
```

2. (30 points) Write a function named processPatients() that loads patient objects from data in a file and returns a list of Patient objects. The data in the file is in string format, you must covert height, age, weight etc to the appropriate type before initialiing the Patient class (see the Animal video from Week 2 for hint how to load from a file).

Patients file contents:



Usage:

```
>>> patients=processPatients('patients.txt')
>>> for p in patients:
        print(str(p))


Bob Jones is 20 years old.  The patient is Overweight and their temp is Normal.
Jessica Williams is 30 years old.  The patient is Normal Weight and their temp is High.
James Brown is 70 years old.  The patient is Normal Weight and their temp is High.
Lucy Fry is 22 years old.  The patient is Overweight and their temp is Low.
>>> for p in patients:
        print(p.getBMIDescription())


Overweight
Normal Weight
Normal Weight
Overweight
>>> for p in patients:
        print(p.tempRange())


Normal
High
High
Low
```

Part 3 (20 points): Inheritance. Complete the Doctor class in the template file. See usage below. Create a Specialist class that inherits from Doctor. A template for Specialist has not been provided. Use the screenshots below for how to implement. You must reuse the Doctor constructor from Specialist.

```
>>> d=Doctor('Steven Strange','NY')
>>> str(d)
'Dr. Steven Strange is affilated with NY hospital.'
>>> d
Doctor('Steven Strange','NY')
>>> s=Specialist('Bruce Banner','Gamma Institute','Gamma Rays')
>>> str(s)
'Dr. Bruce Banner is affilated with Gamma Institute hospital. Their speciality is Gamma Rays.'
>>> s
Specialist('Bruce Banner','Gamma Institute','Gamma Rays')
```

Submitting the assignment

You must submit the assignment using the assignment 2 dropbox on the D2L site. Submit a Python file (csc242hw2.py) with your implementation in it and comments describing your collaboration status. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The assignment is worth 100 points. Any student who does not submit comments in the Python file describing the contributions of each team member or indicating that he/she worked alone will earn a 0 on the assignment.