CSC 242: Introduction to Computer Science II

Assignment 3

Due Tuesday, April 21st, 11:59 pm

Reading

Review Chapter 8 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

Logistics

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 lab assistant, 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 comments 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.

Assignment

Begin the assignment by downloading the template **csc242hw3.py** found on the D2L site. You will complete the classes found there. **You must include appropriate doc strings** (e.g. strings that appear on the line following the class or method header) for the classes and their methods that clearly and concisely describe what the class and methods are doing. A submission without doc strings will not earn full credit.

Do not define any instance variables (e.g. self.x for some variable x) in the classes below other than the ones specified in the assignment description. If you believe you can't write a solution without defining extra instance variables, please ask so that I can give you suggestions. Extra local variables (e.g. x for some variable x) are fine.

PROBLEM 1 (30 POINTS):

Let's get warmed up. Implement the Order object methods. Make sure the output of each method is exactly as shown on the screenshots.

```
>>> o=Order()
>>> o.calculateTotal()
0
>>> str(o)
'::0:0:0'
>>> o=Order('Tech Conference','12/13/2018', 50.00,1000, .09)
>>> str(o)
'Tech Conference:12/13/2018:50.0:1000:0.09'
>>> repr(o)
"Order('Tech Conference','12/13/2018',50.0,1000,0.09)"
>>> copy=eval(repr(o))
>>> str(copy)
'Tech Conference:12/13/2018:50.0:1000:0.09'
>>> copy.getName()
'Tech Conference'
>>> copy.getDate()
'12/13/2018'
>>> copy.getTotalGuests()
1000
>>> copy.getCostPerPerson()
50.0
>>> copy.getTax()
0.09
>>> copy.calculateTotal()
54500.0
>>>
```

PROBLEM 2 (30 POINTS):

Write a class named OrderManager that manages multiple orders. The doc strings in the file describe what each method does. Here are screenshots of the applications behavior.

The constructor takes in the filename used by the application for part 3.

```
>>> o1=Order('Breakfast','1/21/2020',20,10,.25)
>>> o2=Order('Lunch','1/21/2020',25,30,.10)
>>> o3=Order('Dinner','1/21/2020',30,15,.10)
>>> o1.calculateTotal()
250.0
>>> o2.calculateTotal()
825.0
>>> o3.calculateTotal()
495.0
>>> manager=OrderManager()
>>> manager.addOrder(o1)
True
>>> manager.addOrder(o2)
True
>>> manager.addOrder(o3)
True
>>> manager.getOrdersTotal()
1570.0
>>> manager.getCountOfOrders()
3
>>> manager.getAverageCost()
523.3333333333334
>>> manager.getAllEventNames()
['Breakfast', 'Lunch', 'Dinner']
>>> l=manager.getMostExpensiveOrder()
>>> l
Order('Lunch','1/21/2020',25,30,0.1)
```

```
>>> for order in manager:
        print(order.getDate())


1/21/2020
1/21/2020
1/21/2020
>>> for order in manager:
        print(order.getName())


Breakfast
Lunch
Dinner
```

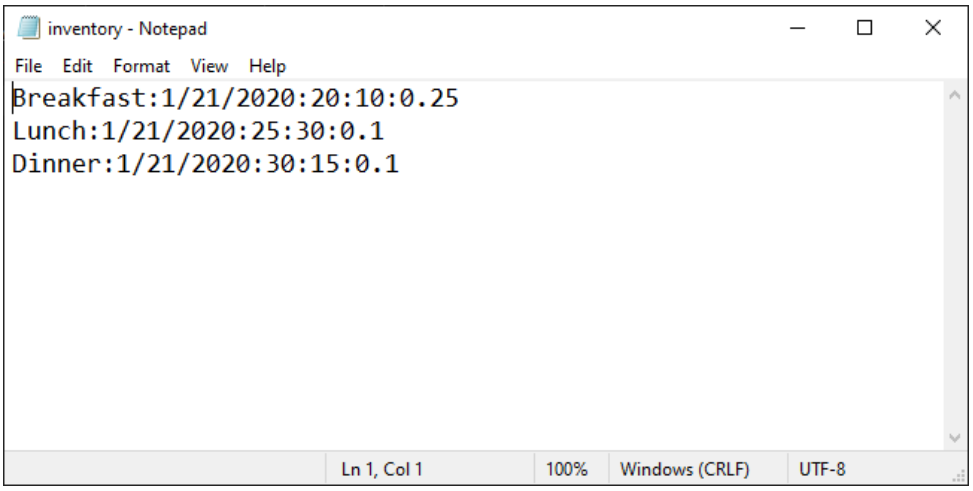
PROBLEM 3 (40 POINTS):

Add methods loadOrders and writeOrdersToFile to OrderManager that read and write orders to a file. See docstrings for more information. You need to use the filename passed in via the constructor to OrderManager. The screenshot below reflects using the same data from the screenshots in part 2.

>>> manager.writeOrdersToFile()

True

Contents of file:



Now we create a new instance of the manager and load the contents into memory:

```
>>> manager2=OrderManager()
>>> manager2.loadOrders()
True
>>> for order in manager2:
        print(str(order))


Breakfast:1/21/2020:20.0:10:0.25
Lunch:1/21/2020:25.0:30:0.1
Dinner:1/21/2020:30.0:15:0.1
>>> manager2.addOrder(Order('test','1/22/2020',100,100,.10))
True
>>> manager.getOrdersTotal()
1570.0
>>> manager2.getOrdersTotal()
12570.0
>>> manager.getAllEventNames()
['Breakfast', 'Lunch', 'Dinner']
>>> manager2.getAllEventNames()
['Breakfast', 'Lunch', 'Dinner', 'test']
```

Submitting the assignment

You must submit the assignment using the assignment 3 dropbox on the D2L site. Submit a Python file (csc242hw3.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.