CMPSC-132: Programming and Computation II
Spring 2019

Lab #5

Due Date: 02/08/2019, 11:59PM

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the LAB5 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class.
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -1 point deduction)
- Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.

Goal:

[5 pts] Write the class *SodaMachine* that will represent a typical soda vending machine (product name, price). An instance of *SodaMachine* has access to three methods, *purchase*, *deposit* and *restock* that will describe its interaction with the user **returning** strings. *Tip*: use the string *format* method

```
>>> m = SodaMachine('Coke', 10)
>>> m.purchase()
'Product out of stock'
>>> m.restock(2)
'Current soda stock: 2'
>>> m.purchase()
'Please deposit $10'
>>> m.deposit(7)
'Balance: $7'
>>> m.purchase()
'Please deposit $3'
>>> m.deposit(5)
'Balance: $12'
>>> m.purchase()
'Coke dispensed, take your $2'
>>> m.deposit(10)
'Balance: $10'
>>> m.purchase()
'Coke dispensed'
>>> m.deposit(15)
'Sorry, out of stock. Take your $15 back'
>>> x = SodaMachine('Dr. Pepper', 8)
>>> x.restock(1)
'Current soda stock: 1'
```

```
>>> x.deposit(8)
'Balance: $8'
>>> x.purchase()
'Dr. Pepper dispensed'
```

- Quotes mean method returned a string, no need to append them in your code
- Return output with the sentences provided. Solution is not case or space sensitive, which means Balance: $10 is the same as balance: $10, but is not the same as Balance= $10

[5 pts] Write the class *Line* that stores the coordinates of two points in a line and provides the distance between the two points and the slope of the line using the **property methods** (no attributes) called *distance* and *slope*. Unless the slope is equal to infinity, both methods must **return** the value as float.

Tip: https://www.pdesas.org/ContentWeb/Content/Content/21083/Lesson%20Plan

- Coordinates must be provided as tuples when creating your class instances
- **Use the round method to format your output to 3 decimals** [round(ouput_value, 3)]**.** Incorrect format will result in -1 pt from your score

EXAMPLES:

```
>>> line1=Line((8,3),(0,-4))  #Coordinates provided as tuple
>>> line1.distance
10.63
>>> line1.slope
0.875
>>> line2=Line((-7,-9),(1,5.6))
>>> line2.distance
16.648
>>> line2.slope
1.825
>>> line3=Line((2,6),(2,3))
>>> line3.distance
3.0
>>> line3.slope
'Infinity'
```

- Quotes mean method returned a string, no need to append them in your code

Deliverables:

- Submit your code with the file name LAB5.py to the Lab5 GradeScope assignment before the due date