CSC1001: Introduction to Computer Science
Programming Methodology
Assignment 3

Assignment description:
This assignment will be worth 9% of the final grade.

You should write your code for each question in a .py file (please name it using the question name, e.g. q1.py). Please pack all your .py files into a single .zip file, name it using your student ID (e.g. if your student ID is 123456, then the file should be named as 123456.zip), and then submit the .zip file via Moodle.

Please also write a text file, which provide the details about how to run your code for each question. The text file should be included in the .zip file as well.

Please note that, the teaching assistant may ask you to explain the meaning of your program, to ensure that the codes are indeed written by yourself. Please also note that we may check whether your program is too similar to your fellow students' code using Moodle.

This assignment is due on 5:00PM, 15 Apr (Sunday). For each day of late submission, you will lose 10% of your mark in this assignment. If you submit more than three days later than the deadline, you will receive zero in this assignment.

Question 1 (20% of this assignment):
Write a Python class, Flower, that has three instance variables of type str, int, and float, that respectively represent the name of the flower, its number of petals, and its price. Your class must include an initializer that initializes each variable to an appropriate value, and your class should include methods for setting the value of each type, and retrieving the value of each type. Your program should be robust enough to handle possible inappropriate inputs.

Question 2 (40% of this assignment):
Write a Python class that inputs a polynomial in standard algebraic notation and outputs the first derivative of that polynomial. Both the inputted polynomial and its derivative should be represented as strings.

For example, when the inputted polynomial is $'2*x^3+3*x^2+5*x+1'$, the output of your program should be $'6*x^2+6*x+5'$.

Note: (1) The inputted polynomial will contain only one variable, and the variable is not

necessarily 'x'; (2) In the inputted polynomial, the terms are not necessarily arranged in descending or ascending orders.

Question 3 (*40% of this assignment*):
Write a Python class to simulate an ecosystem containing two types of creatures, bears and fish. The ecosystem consists of a river, which is modeled as a relatively large list. Each element of the list should be a Bear object, a Fish object, or None. In each time step, based on a random process, each animal either attempts to move into an adjacent list location or stay where it is. If two animals of the same type are about to collide in the same cell, then they stay where they are, but they create a new instance of that type of animal, which is placed in a random empty (i.e., previously None) location in the list. If a bear and a fish collide, however, then the fish dies (i.e., it disappears).

Write an initializer for the ecosystem class, the initializer should allow the user to assign the initial values of the river length, the number of fishes and the number of bears. Before the simulation, fishes and bears should be allocated randomly into the river. The ecosystem class should also contain a simulation() method, which will simulate the next N steps of the random moving process. N should be inputted by the user. In each step of your simulation, all animals in the river should try to take some random moves.

For example, assume that before the simulation, the initial state of the river is:
$$\text{NNFNNNBNBNN}$$
In which, 'F', 'B' and 'N' denote fish, bear and empty location respectively. Assume that in the first step of simulation, the first fish will move to the left, the first bear will move to the right, and the second bear will remain still. Then after the first step, the state of the river is:
$$\text{NFNNNNNBBNN}$$

To generate random numbers in Python, you should import the random() function by using the following statement:

```
from random import random
```

By assigning the return of the random() function to a variable, you will get a random floating point number in the range of [0, 1]. The following code is an example of using the random() function:

```
>>> x=random()
>>> print(x)
0.8303929183861047
```