CSCI 331: Object Oriented Software Development

Lab #4 (100 points)

Due: Wednesday, March 15 by 23:59

Directions: The following lab will help you explore how to write Python programs using object-oriented software development concepts. Please label your programs `q<num>.py`, where `num` is the question number. For example, your solution to the first question, will be stored in the file `q1.py`.

Your answers should be based on the content and procedures covered in class, and not on other implementations or descriptions that can be found online.

Your programs should be driven by logic. You will not receive credit if you use statements to jump out of a loop: break, continue, etc.

Take your time and make sure you understand everything in this document before getting started. Make sure your programs match the output EXACTLY as given for each question. This is important, as one of the keys to being a good programmer is attention to details. Finally, don't leave things to the last minute.

1 Questions

1. Write a program (called `q1.py`) that extends an abstract base class. Python's **collections** module has some concrete classes that derive from abstract base classes; these can, of course, be further derived.

   You will be extending Python's **UserList** class: a wrapper around list objects for easier list subclassing. More specifically, you will write the code for the methods in the `Lab4` class. These methods must directly change the the object, and not make a copy of it:

   - **repeat(n)**: Repeat the existing object **n** times.
   - **add(x)**: Adds **x** to the end of the object.
   - **remove(m, n)**: Removes all elements between indices **m** and **n** (inclusive) in the object.
   - **concat(x)**: concatenates the object with x (which can be another object derived from a **UserList** class).
   - **depth(x)**: Assuming that the object contains other nested objects, Should return the depth of the deepest **UserList**.

   For example, this is a sample output from loading **Lab4** to the interactive interpreter:

```
>>> test = Lab4([0])
>>> print(test)
[0]
>>>
>>> test.repeat(4)
>>> print(test)
[0, 0, 0, 0, 0]
>>>
>>> test = Lab4([0, 1, 2])
>>> test.repeat(2)
>>> print(test)
[0, 1, 2, 0, 1, 2, 0, 1, 2]
>>>
>>> test.add(45)
>>> print(test)
[0, 1, 2, 0, 1, 2, 0, 1, 2, 45]
>>>
>>> test.remove(7, 7)
>>> print(test)
[0, 1, 2, 0, 1, 2, 0, 2, 45]
>>>
>>> test.remove(3, 5)
>>> print(test)
[0, 1, 2, 0, 2, 45]
>>>
```

```
>>> test.concat(Lab4([12, 13]))
>>> print(test)
[0, 1, 2, 0, 2, 45, 12, 13]
>>>
>>> test.concat(Lab4([43]))
>>> print(test)
[0, 1, 2, 0, 2, 45, 12, 13, 43]
>>>
>>> print(test.depth(test.data))
1
>>>
>>> test.add(Lab4([0, 0]))
>>> print(test)
[0, 1, 2, 0, 2, 45, 12, 13, 43, [0, 0]]
>>> print(test.depth(test.data))
2
```

Example #1. This is a sample execution of the program, and it shows that the values inserted in the object are randomized. The initial state of the object is shown on line 3.

```
1  python q1.py
2  Test 0 - initial state:
3  [18, 73, 37, 61]
4  Test 1 - repeat():
5  [18, 73, 37, 61, 18, 73, 37, 61, 18, 73, 37, 61]
6  Test 2 - add():
7  [18, 73, 37, 61, 18, 73, 37, 61, 18, 73, 37, 61, 39, 71]
8  Test 3 - remove():
9  [18, 73, 37, 61, 18, 73, 37, 61, 39, 71]
10 Test 4 - concat():
11 [18, 73, 37, 61, 18, 73, 37, 61, 39, 71, 99, 43, 71]
12 Test 5 - depth():
13 [18, 73, 37, 61, 18, 73, 37, 61, 39, 71, 99, 43, 71, [13, 10], [[26], 22]]
14 depth: 3
15 Elapsed time is 0.000173 seconds.
16 **************
```

Example #2. This is another sample execution of the program, and it shows that the values inserted in the object are randomized. The initial state of the object is shown on line 3.

```
1  python q1.py
2  Test 0 - initial state:
3  [4, 22, 26, 80]
4  Test 1 - repeat():
5  [4, 22, 26, 80, 4, 22, 26, 80, 4, 22, 26, 80]
6  Test 2 - add():
7  [4, 22, 26, 80, 4, 22, 26, 80, 4, 22, 26, 80, 3, 73]
8  Test 3 - remove():
9  [4, 22, 26, 80, 4, 22, 26, 80, 3, 73]
10 Test 4 - concat():
11 [4, 22, 26, 80, 4, 22, 26, 80, 3, 73, 93, 80, 73]
12 Test 5 - depth():
13 [4, 22, 26, 80, 4, 22, 26, 80, 3, 73, 93, 80, 73, [24, 81], [[56], 24]]
14 depth: 3
15 Elapsed time is 0.000748 seconds.
16 **************
```

2 Important Links

- https://docs.python.org/3/library/abc.html
- https://docs.python.org/3/library/collections.html
- https://docs.python.org/3/library/collections.abc.html

3 Submitting Your Assignment

Once you have completed your program, submit it (q1.py) in VIU Learn in its corresponding Assignment page.