CSCI 331: Object Oriented Software Development

Lab #2 (100 points)

Due: Wednesday, February 4 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 parses XML files. More specifically, you will be parsing XML files that contain the lines from Shakespeare's plays. You will be using **Beautiful Soup** and **lxml** for this task. Beautiful Soup is a Python library for pulling data out of HTML and XML files. lxml is the most feature-rich and easy-to-use library for processing XML and HTML in the Python language.

   You will write the code for the methods in the `LabTwo` class:

   - **readFile()**: takes a string corresponding to a file name as a parameter, and returns a Beautiful Soup object. This object will be accessed by the other methods in the class.
   - **getPlayName()**: takes a Beautiful Soup object as a parameter, and returns a string corresponding to the name of the play that is referenced in the file.
   - **getCharacters()**: takes a Beautiful Soup object as a parameter, and returns a Python dictionary. Each key in the dictionary corresponds to a character, the values are a list of items corresponding to the scenes and acts where the given character appears.

   For example, a portion of the file `playtext/tempest.xml` is shown below.

```
1  <play name="Tempest">
2    <div type="act" n="1">
3      <div type="scene" n="1">
4        <sp who="#(stage_directions)">
5          <lb n="4">[Enter a Master and a Boatswain]</lb>
6        </sp>
7        <sp who="#Master">
8          <lb n="5">Boatswain!</lb>
9        </sp>
10       <sp who="#Boatswain">
11         <lb n="6">Here, master: what cheer?</lb>
12       </sp>
13       <sp who="#Master">
14         <lb n="7">Good, speak to the mariners: fall to't, yarely,
15 or we run ourselves aground: bestir, bestir.</lb>
```

   - Line 1 contains the name of the play
   - Line 2 contains the act number
   - Line 3 contains the scene number
   - Lines 4, 7, 10, and 13 contain the character names

Example #1. At the prompt, the user provides `playtext/tempest.xml` as a command line argument (line 1). The characters are reported on lines 5-24.

```
1  python q1.py playtext/tempest.xml
2  Reading file playtext/tempest.xml ...
3  Play name: Tempest
4  Characters:
5  (stage_directions): 1.1 1.2 2.1 2.2 3.1 3.2 3.3 4.1 5.1
6  Master: 1.1
7  Boatswain: 1.1 5.1
8  Alonso: 1.1 2.1 3.3 5.1
9  Antonio: 1.1 2.1 3.3 5.1
10 Gonzalo: 1.1 2.1 3.3 5.1
11 Sebastian: 1.1 2.1 3.3 5.1
12 Mariners: 1.1
13 Miranda: 1.2 3.1 4.1 5.1
14 Prospero: 1.2 3.1 3.3 4.1 5.1
15 Ariel: 1.2 2.1 3.2 3.3 4.1 5.1
16 Caliban: 1.2 2.2 3.2 4.1 5.1
17 Ferdinand: 1.2 3.1 4.1 5.1
18 Adrian: 2.1 3.3
19 Francisco: 2.1 3.3
20 Trinculo: 2.2 3.2 4.1 5.1
21 Stephano: 2.2 3.2 4.1 5.1
22 Iris: 4.1
23 Ceres: 4.1
24 Juno: 4.1
25 **********
26 Elapsed time is 0.047310 seconds.
```

Example #2. At the prompt, the user provides `playtext/midsummer.xml` as a command line argument (line 1). The characters are reported on lines 5-28.

```
1  python q1.py playtext/midsummer.xml
2  Reading file playtext/midsummer.xml ...
3  Play name: Midsummer Night's Dream
4  Characters:
5  (stage_directions): 1.1 1.2 2.1 2.2 3.1 3.2 4.1 4.2 5.1
6  Theseus: 1.1 4.1 5.1
7  Hippolyta: 1.1 4.1 5.1
8  Egeus: 1.1 4.1
9  Hermia: 1.1 2.2 3.2 4.1
10 Demetrius: 1.1 2.1 2.2 3.2 4.1 5.1
11 Lysander: 1.1 2.2 3.2 4.1 5.1
12 Helena: 1.1 2.1 2.2 3.2 4.1
13 Quince: 1.2 3.1 4.2 5.1
14 Bottom: 1.2 3.1 4.1 4.2 5.1
15 Flute: 1.2 3.1 4.2 5.1
16 Starveling: 1.2 3.1 4.2 5.1
17 Snout: 1.2 3.1 5.1
18 Snug: 1.2 4.2 5.1
19 All: 1.2 3.1
20 Puck: 2.1 2.2 3.1 3.2 4.1 5.1
21 Fairy: 2.1 2.2
22 Oberon: 2.1 2.2 3.2 4.1 5.1
23 Titania: 2.1 2.2 3.1 4.1 5.1
24 Peaseblossom: 3.1 4.1
25 Cobweb: 3.1 4.1
26 Moth: 3.1
27 Mustardseed: 3.1 4.1
28 Philostrate: 5.1
29 **********
30 Elapsed time is 0.040973 seconds.
```

Example #3. At the prompt, the user provides `playtext/foo.txt` as a command line argument (line 1). This argument corresponds to a file that does not exist. The program raises an exception and determines that an error occurred (line 2).

```
1  python q1.py playtext/foo.txt
2  ERROR: cannot open playtext/foo.txt
3  Elapsed time is 0.000925 seconds.
```

2 Important Links

- Beautiful Soup Documentation
- lxml - XML and HTML with Python
- Open Source Shakespeare

3 Submitting Your Assignment

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