CSCI 331: Object Oriented Software Development

Lab #5 (100 points)

Due: Wednesday, April 29 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 the elements of a web page, and retrieves specific documents linked from it. There are many things that can go wrong with these tasks, so you will need to use multiple exceptions.

   Here are the steps you will need to complete this lab.:

   - Create a **requests.get()** object using two parameters:
     url: provided as a command line argument, and
     headers: already defined in the source code.
   - If the HTTP status code of the **requests** object is equal to **200**, create a **BeautifulSoup** object from the contents of the web page using the built-in **html.parser**. Otherwise, output an error message and the HTTP status code for the retrieved page.
     For example, for a 404 HTTP status code: **Error retrieving page: 404**.
   - Using the newly created **BeautifulSoup** object, find all the hyperlinks for **.docx** files.
     Then create a random subset of these links using the **random.choices()** method. The **numDocs** variable specifies the size of this subset.
   - Create another **requests** object, and iteratively save the random subset of **.docx** documents into the **docDir** folder.
     For example: if the *.docx* file is named:
     2021.10.06-ENCUENTRO-MINERIA.docx, it will be saved as:
     docs/2021.10.06-ENCUENTRO-MINERIA.docx
   - Finally, extract the text contents of the saved documents using the **docxpy** library. Save the corresponding text for each of the **.docx** documents as a text file in the **txtDir** folder.
     For example, if the *.docx* file is named:
     docx/2021.10.06-ENCUENTRO-MINERIA.docx,
     its corresponding *.txt* file would be saved as:
     txts/2021.10.06-ENCUENTRO-MINERIA.txt.
   - Remember to be thorough and close each **requests** and **file** object. Keep in mind that your program must recover cleanly from errors, which are very common in network connections and file operations.

Example #1. At the prompt, the user provides a URL as a command line argument (line 1). No errors occurred during execution.

```
1  python q1.py https://www.presidencia.gob.ec/discursos/
2  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
      ↪ downloads/2022/12/2022.12.15-EQUIPOS-POLICIA.docx
3  txt file saved: txts/2022.12.15-EQUIPOS-POLICIA.txt
4  **********
5  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
      ↪ downloads/2022/02/2022-02-16-DONACION-CHINA-LA-GASCA.docx
6  txt file saved: txts/2022-02-16-DONACION-CHINA-LA-GASCA.txt
7  **********
8  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
      ↪ downloads/2022/07/2022.07.26-ENCONTREMONOS-INVERSIONISTAS.
      ↪ docx
9  txt file saved: txts/2022.07.26-ENCONTREMONOS-INVERSIONISTAS.txt
10 **********
11 Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
      ↪ downloads/2022/12/2022.12.06-FUNDACION-QUITO.docx
12 txt file saved: txts/2022.12.06-FUNDACION-QUITO.txt
13 **********
14 Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
      ↪ downloads/2022/08/2022.08.04-VIVIENDAS-SAN-SEBASTIAN.docx
15 txt file saved: txts/2022.08.04-VIVIENDAS-SAN-SEBASTIAN.txt
16 **********
17 closing connection....
```

Example #2. At the prompt, the user provides a URL as a command line argument (line 1). Errors occurred when processing files 3 and 4, so errors are displayed on lines 9 and 12 respectively.

```
1  python q1.py https://www.presidencia.gob.ec/discursos/
2  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
       ↪ downloads/2022/11/2022.11.23-HEROES-CONTRA-DELINCUENCIA.
       ↪ docx
3  txt file saved: txts/2022.11.23-HEROES-CONTRA-DELINCUENCIA.txt
4  **************
5  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
       ↪ downloads/2023/01/2023.01.26-VUELOS-MANTA-PANAMA.docx
6  txt file saved: txts/2023.01.26-VUELOS-MANTA-PANAMA.txt
7  **************
8  Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
       ↪ downloads/2022/07/2022.07.14-VIVIENDAS-HUAQUILLAS.docx
9  ERROR - unable to save text file
10 **************
11 Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
       ↪ downloads/2022/12/2022.12.08-CASAS.YANTZAZA.docx
12 ERROR - unable to save text file
13 **************
14 Retrieving: https://www.presidencia.gob.ec/wp-content/uploads/
       ↪ downloads/2021/06/2021.05.31-CAMBIO-DE-MANDO-CARONDELET.
       ↪ docx
15 txt file saved: txts/2021.05.31-CAMBIO-DE-MANDO-CARONDELET.txt
16 **************
17 closing connection....
```

Example #3. At the prompt, the user provides a URL as a command line argument (line 1). The URL points to a page that does not exist, so the **requests** object returns a 404 status code. The error message is shown on line 2.

```
1  python q1.py https://www.presidencia.gob.ec/foo/
2  Error retrieving page: 404
3  closing connection....
```

2 Important Links

- https://requests.readthedocs.io/en/latest/user/quickstart/
- https://pypi.org/project/user-agent/
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- https://docs.python.org/3/library/random.html
- https://pypi.org/project/docxpy/
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

3 Submitting Your Assignment

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