Pair Exercise #4

Fall 2025

Submission: Github repository link. Include the name of your partner.
Due: 11/6/2025, 6:59 PM
Assignment type: Pair
Points: 40

INSTRUCTIONS

Create code as outlined below in a file named **pe4.py** and push it to a github repository (you can use the same repo as previous pair exercises if you are paired with the same partner, something like **303_Fall_25**). Your file will have code that addresses Section A as well as Section B.

The wikipedia package enables access to and extraction of Wikipedia content using Python. More details on how the package works, including syntax, can be found here:

- https://www.thepythoncode.com/article/access-wikipedia-python.

You can install the package in your virtual environment using pip as you would any other package: `pip install wikipedia` Use this package as described below.

A. Sequentially download wikipedia content

1. Use the **wikipedia.search** method to return a list of topics related to 'generative artificial intelligence'.
2. Iterate over the topics returned in #1 above using a for loop. For each topic, do the following:
    - assign the page contents to a variable named *page* using the **wikipedia.page** method. Be sure to use **auto_suggest=False** when using this method
    - assign the page title to a variable (using **page.title**)
    - retrieve the references for that page (using **page.references**)
    - write the references (with each reference on its own line) to a .txt file where the name of the file is the title of the topic. For example, the topic "Music and artificial intelligence" would be written to a file named *Music and artificial intelligence.txt*. You'll need to pay attention to the variable type returned by *page.references*, as only certain objects can be written to .txt files using the OBJECT.write method. You may also need to manipulate what is returned to ensure each link is on its own line.
3. Print to the console the amount of time it took the above code to execute, using time.perf_counter() (to finish step 2 for ALL topics in the list)

B. Concurrently download wikipedia content

1. Use the **wikipedia.search** method to return a list of topics related to 'generative artificial intelligence'.
2. Create a function def **wiki_dl_and_save(topic):** that:
    - retrieves the wikipedia page for the topic

    - gets the title and the references for the topic
    - creates a .txt file where the name of the file is the title of the topic
    - writes the references to the file created in the preceding step (each reference should be on its own line)
3. Use the **ThreadPoolExecutor** from the *concurrent.futures* library to execute concurrently the function defined in step 2.
    - you can call the .map() method of the ThreadPoolExecutor object with the function name and list of topics to assist with this step (the map method will execute a specified function for each object in an iterable that is passed in)
4. Print to the console the amount of time it took the above code to execute, using time.perf_counter()

Additional information

The only function defined by name is the **wiki_dl_and_save(topic)** function. You are free to write additional helper functions or abstracted functions if it makes more sense for you.

Reminder: As always, do not use AI or coding assistant tools to assist with code generation. All of the information you need is contained in this document, the linked material, and/or the lecture materials (you may want to refer back to earlier material, for example module 6 slides covering writing to files).