FILE SEARCH

PURPOSE

In this exercise, file IO is explored using Python. The user is prompted for a file path to search, and a word to search for, and then a file is opened to search for the user-entered phrase. The results are then printed to console and written to file.

OBJECTIVES

After completing this exercise, you should be able to:

- Open a file for reading using Python
- Write to a file using Python
- Check if a file exists using Python

PROCEDURE

PREPARE SUBMISSION FILE

1. Create a copy of the submission template called COMP6060**INIT**Lab7.docx where **INIT** is replaced with your own initials. So if your name is John Smith, the document will be called COMP6060**JS**Lab7.docx

PREPARE PYTHON FILE

1. Create a Python file called COMP6060**INIT**Lab7.py where **INIT** is replaced with your own initials. So if your name is John Smith, the document will be called COMP6060**JS**Lab7.py
   Print out the following to the console, replacing NAME with your name:
   ```
   Welcome to NAME’s file search utility!
   ```
2. In the same folder, create a file called data.txt with the following text:
   ```
   This is the first line
   This is the second line
   and third...
   And fourth!!
   ```

Open File

1. Using the `input()` function, prompt the user for a file path to open. Store the file path in a variable called `file_path`
    a. Use the following prompt:
    `Enter the file path to search:`
2. Check if the file path provided by the user exists in the filesystem. To do that:
    a. Add the following import statement at the very top of the Python file: `from os import path`
    b. Exit the program if the file does not exist:
    ```
    if not path.exists(file_path):
        print(f"The file {file_path} does not exist… exiting")
        exit(-1)
    ```
3. Open the file in `filepath` in read mode. Assign the file handle to a variable called `file_to_search`

Search For Word

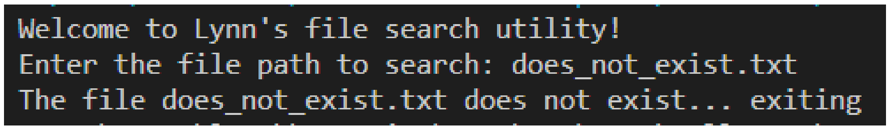
1. Prompt the user for a single word to search. Store the word in a variable called `word_to_find`
    a. Use the following prompt:
    `Enter a single word (no spaces) to search the file:`
2. Create a variable called `counter`, and assign it the value 0. This will hold the number of instances of the word we might find in the file
3. Using nested ranged for loops and the `split()` function, read the file word by word
    a. Inside the inner ranged for loop, check if the word is equal to `word_to_find`. If it is, increment counter
4. Once the for loops are complete, close the file `file_to_search`

Output Results

1. Open a file called "log.txt" in write mode. This file will not exist initially, and will be created in the current working directory. Store the file handle in a variable called `file_to_write`
2. Build a formatted string with the following format:
    word was found in *filename*: *count* times
3. Write the formatted string to the file `file_to_write`
4. Print the formatted string to console
5. Close the file `file_to_write`

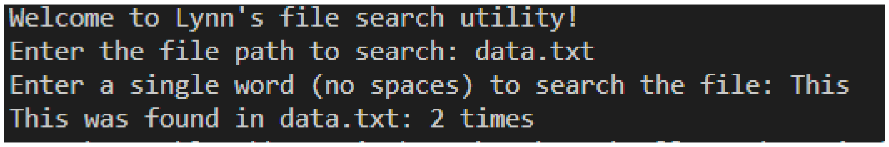
Expected Output

When the file does not exist



```
Welcome to Lynn's file search utility!
Enter the file path to search: does_not_exist.txt
The file does_not_exist.txt does not exist... exiting
```

When the file exists



```
Welcome to Lynn's file search utility!
Enter the file path to search: data.txt
Enter a single word (no spaces) to search the file: This
This was found in data.txt: 2 times
```

Show results to Instructor.

Student Name: _______________________ Instructor: _______________________

Date: _______________________