Tuesday Nov 5, 2024

In-class Handout

COSC 101C Intro to Computing I

Prof. Forrest Davis

Name:

Discuss and complete the following questions with the person nearest you. You **may** be asked to share your thoughts with the class.

1. Modify the code below such that **read_data** function returns None, instead of causing the program to crash, if the file does not exist.

```
def read_data(filename:str) -> str:
    """Read data from a file
    Parameters:
        filename (str): Name of file
    Returns:
        data (str): Contents of the file
    """
    infile = open(filename, 'r')
    data = infile.read()
    infile.close()
    return data

def main() -> None:
    data = read_data('does-not-exist.txt')
    print(data)

main()
```

```
def read_data(filename:str) -> str:
    """Read data from a file
    Parameters:
        filename (str): Name of file
    Returns:
        data (str): Contents of the file or none if file
                    does not exist
    """
    try:
        infile = open(filename, 'r')
        data = infile.read()
        infile.close()
        return data
    except FileNotFoundError:
        return None

def main() -> None:
    data = read_data('does-not-exist.txt')
    print(data)

main()
```

2. Write a function called **odd_lines** that takes the name of a file and returns a list containing the 1st, 3rd, 5th, etc.

lines of the file.

Assume file1.txt contains the following data

```
First
Second
Third
Fourth
Fifth
Sixth
Seventh
```

```
def createFile() -> None:
    """ Creates file1.txt with the contents described above"""
    with open('file1.txt', 'w') as f:
        for expression in ['First', 'Second',
                           'Third', 'Fourth', 'Fifth',
                           'Sixth', 'Seventh']:
            f.write(f"{expression}\n")
def odd_lines(filename: str) -> list:
    """ Construct a list containing the 1st, 3rd, 5th,
        etc. lines of the file
    Parameters:
        filename (str): Name of file
    Returns:
        lines (list): List of lines in file

    >>> odd_lines('file1.txt')
    ['First\\n', 'Third\\n', 'Fifth\\n', 'Seventh\\n']
    """
    lines = []
    with open(filename, 'r') as infile:
        line = infile.readline() # Read first line
        while line: # Repeat until end of file
            lines = lines + [line]
            line = infile.readline() # Skip a line
            line = infile.readline()

    return lines

createFile()
odd_lines('file1.txt')
import doctest
doctest.testmod()
```

3. Write a function to extract all the `sentences` from the following data (imagine it's in a file):

   speaker_role,sentence,adjective

   Target_Child,and an angry beaver or something came out and bit him in the nose,angry

   Target_Child,but because I love my dog so much I wasn't too angry at him,angry

   Target_Child,um they're quite happy and surprised to see them,happy

   Target_Child,and in the last picture the dog is pretty happy bounding about,happy

   Target_Child,and then Tom doesn't look too happy when he's licking his face,happy

   Target_Child,and they went deeper into the woods and met an angry owl,angry

Target_Child,so he's maybe a little bit jealous,jealous

Target_Child,um they're rude unhappy for no good reason,happy

Target_Child,and then our form tutor just gets angry angry and shouts at us all and then moans at us all and like,angry

Target_Child,and she gets mad because I'm I don't have the same lifestyle as her,mad

```python
def extractSentences(fname: str) -> list:
    sentences = []
    with open(fname, 'r') as f:
        # remove header
        f.readline()
        for line in f:
            sentence = line.split(',')[-2]
            sentences.append(sentence)
    return sentences
```