HLT – Spring 2021

Homework 1: Text Processing with Python

**Objective**: Use Python to process text.

**Turn in:** your Python .py file

**Scenario**: An employee file has been created with an obsolete system. Your task is to read in the file, process the text to be more standardized as described below, create an object for each person with corrections from the user, and output each person’s information.

**Input**: The input file (data.csv) looks like this.

Last,First,Middle Initial,ID,Office phone

Smith,Smitty,S,WH1234,5557771212

WILLIAMS,WITTY,W,S4454,555-877.4321

Luka,Luka,L,OF4321,555.888.3456

jason,jake,,WH409,555 777 2094

Krishna,krishna,k,SA9384,555 888 0093

**Instructions**:

A screenshot of a cell phone

Description automatically generated

1. Download the csv file from Piazza and place it in a folder named data within the same folder as your Python program. Here is a screen shot of the folder structure to make this clearer. Also, the grader would prefer that your uploads include your netid. For this homework, you will just upload the .py file.
2. The user needs to specify the relative path ‘data/data.csv’ in a sysarg. If the user does not specify a sysarg, print an error message and end the program. Read the file, making sure your program will work on either a Windows or Mac/Unix. See the Paths Demo in the Xtra folder of the GitHub: <https://github.com/kjmazidi/NLP>
3. Define a person class with fields last, first, mi, id, and phone. In addition to the init method, create a display() method to output fields as shown in the sample run below.
4. Create a function to process the input file. Get rid of the first line which is just the heading line. For the remaining lines:
   1. split on comma to get the fields as text variables
   2. modify last name and first name to be in Capital Case, if necessary
   3. modify middle initial to be a single upper case letter, if necessary. Use ‘X’ as a middle initial if one is missing.
   4. modify id if necessary, using regex. The id should be 2 letters followed by 4 digits. If an id is not in the correct format, output an error message, and allow the user to re-enter a valild ID. See the sample run below for data corrections.
   5. modify phone number, if necessary, to be in form 999-999-9999. Use regex.
   6. Once the data for a person is correct, create a Person object and save the object to a dict of persons, where id is the key. Check for duplicate id and print an error message if an ID is repeated in the input file.
   7. Return the dict of persons to the main function.
5. In the main function, save the dictionary as a pickle file. Open the pickle file for read, and print each person using the Person display() method to verifiy that the pickle was unpickled correctly. There is a sample pickle notebook in the Xtras folder in the GitHub.

**Grading Rubric:**

|  |  |
| --- | --- |
| Element | Points |
| Appropriate comments, white space, functions | 10 |
| Step 1 | 10 |
| Step 2 | 10 |
| Step 3 | 10 |
| Step 4 | 40 |
| Step 5 | 20 |
| Total | 100 |

**Sample run:**

ID invalid: S4454

ID is two letters followed by 4 digits

Please enter a valid id: SA4454

Phone 555-877.4321 is invalid

Enter phone number in form 123-456-7890

Enter phone number: 555-877-4321

ID invalid: WH409

ID is two letters followed by 4 digits

Please enter a valid id: WH5409

Employee list:

Employee id: WH1234

Smitty S Smith

555-777-1212

Employee id: SA4454

Witty W Williams

555-877-4321

Employee id: OF4321

Luka L Luka

555-888-3456

Employee id: WH5409

Jake X Jason

555-777-2094

Employee id: SA9384

Krishna K Krishna

555-888-0093

Process finished with exit code 0