

### Text Processing with Python

This python program processes texts from a csv file by splitting each string and verifying their validity before displaying a list of employees with their information. First, the user enters the data's file path in the command prompt as an argument, where the program can find the csv file to open and read. The csv file contained a header on the first line that I had to remove, then the file contained [Last, First, Middle Initial, ID, Phone Number] in order, and repeated for 5 times for 5 employees.

	A	B	C	D	E
1	Last	First	Middle Initial	ID	Office phone
2	Smith	Smitty	S	WH1234	5557771212
3	WILLIAMS	WITTY	W	S4454	555-877.4321
4	Luka	Luka	L	OF4321	555.888.3456
5	jason	jake		WH409	555 777 2094
6	Krishna	krishna	k	SA9384	555 888 0093
7					

Figure 1: The initial input data file, as data.csv

The program verifies the last name to make sure only the first letter is capitalized, then verifies the first name for the same format. Then it checks the middle initial to make sure it is a single uppercase letter, unless it does not exist, in which case a letter 'X' will take the spot for. Then the program verifies the ID number for the employees, to make sure that they are all in a format of 2 letters followed by 4 digits. If something is not valid here, the program prompts the user to enter a new value, where the program can then re-verify until it fits the criteria.

Once all input data are valid, they are then split into employees, and each employee will get their name, ID, and phone number, and they get put into a dictionary, and during this process the program also checks that there are no duplicate IDs between any of the employees.

When all of the data processing is completed, the dictionary of employees is saved as a pickle byte file, so it can be loaded later at any time. The program immediately displays the content in the pickle file to the console to show the processed text results clean and formatted.

```
=====Employee list=====
Employee ID   : WH1234
Employee Name : Smitty S Smith
Employee Phone : 555-777-1212

Employee ID   : SH4454
Employee Name : Witty W Williams
Employee Phone : 555-877-4321

Employee ID   : OF4321
Employee Name : Luka L Luka
Employee Phone : 555-888-3456

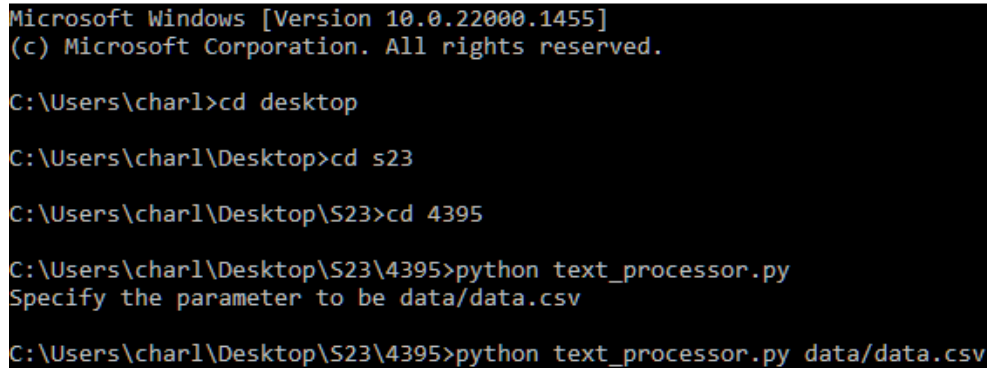
Employee ID   : WH4090
Employee Name : Jake X Jason
Employee Phone : 555-777-2094

Employee ID   : SA9384
Employee Name : Krishna K Krishna
Employee Phone : 555-888-0093
```

Figure 2: Employee ID, Name, and Phone is displayed on the command prompt.

Since the project was done on Windows, Here is how to run it on windows.

1. First you need to save the text\_processor.py file, and create a folder named 'data' and save data.csv file inside the data folder. The 'data' folder must be in the same directory as the text\_processor.py file.
2. Run the command prompt by pressing Win+R and typing 'cmd', then press Enter.
3. Change directory in cmd to locate the program file "text\_processor.py." I saved my python program in the 4395 folder inside S23 on Desktop, so this is how I got there. Another easy way is to right click on the python file, and copy the file path to the folder this program is located at.
4. Once you are in the correct directory, enter: [ python text\_processor.py data/data.csv ] as shown



```
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

C:\Users\charl>cd desktop

C:\Users\charl\Desktop>cd s23

C:\Users\charl\Desktop\S23>cd 4395

C:\Users\charl\Desktop\S23\4395>python text_processor.py
Specify the parameter to be data/data.csv

C:\Users\charl\Desktop\S23\4395>python text_processor.py data/data.csv
```

Figure 3: Locating the program file and running the file with data.csv as sysarg

5. The program will give you an output similar to Figure 2, otherwise follow the program instructions to validate your data.

In my opinion, Python is much easier to process text than other languages like Java or C/C++, because I can easily split between spaces or commas, and the verification steps are not complicated. I remember using compareTo() methods in Java to compare two strings lexicographically, which would not have worked well with trying to verify limitations and boundaries like I did in python using regex. I think Python is powerful and simple enough where I don't have to spend hours looking for a specific way to solve a problem, because it offers many directions.

This assignment was a great review on using python as a script, because for the past few months I've been using jupyter notebooks without system arguments as inputs. I also got to review using regex, and most aspects of the program were not too difficult. I also learned that the input file type does not matter too much, because when I convert the csv file to a text file (txt), all of the spreadsheet cells were converted to commas and I was able to see more clearly what the program is reading as the input.