Name: Bushra Rahman
Class: CS 4395.001
Assignment: NLP Portfolio Component 1

Overview of the Program

The purpose of this program is to read in a CSV file containing employee information and process the lines of text describing each employee. A file called data.csv is read into a list of strings in `main()`, where each string is a line from the file. From there, a function called `process_lines()` splits each string into its respective tokens — last name, first name, middle initial, employee ID, and phone number — and processes them accordingly. Once all of an employee's fields have been processed, a `Person` object is created for them, where each of those tokens corresponds to a field in `Person`. Finally, a dictionary is returned, which contains each `Person` object as the value and its respective employee ID as the key. This dictionary is dumped into a pickle file in `main()`, which then reads it back in and calls the `Person` class's `display()` function to display the employees.

How to Run the Program

To run this program, the user needs to specify the relative path to data.csv as a sysarg. To do this in PyCharm, open the dropdown menu for the .py file and open "Edit Configurations". There, for "Parameters", specify the relative path: `data/data.csv`, where data is the folder containing data.csv.

Strengths and Weaknesses of Text Processing With Python

Text processing with Python is fairly straightforward and simple to implement. The built-in `split()` function in Python makes it easy to tokenize lines of text that aren't necessarily natural language, but rather, simple lists of items, as in this project. If the lines of text in data.csv were more akin to natural English sentences, then Python's `split()` function would be less suited for the job than, say, NLTK's `word_tokenize()` function. However, because the lines of text in this project's input file were simply substrings of data about a person — their names and numbers — standard Python was fully capable of processing it.

What I Learned

This project was a thorough review of Python for me: defining and calling functions, defining and calling a class, and file handling. I felt it was a solid introduction to the world of text processing while also being a refresher on Python, so that I am prepared to handle more complex text processing tasks using NLTK later on in the course.