CSEE5590/490:

Python and Deep Learning Programming

(2018 Fall)

*Python Lab Assignment 1*

Submitted On:

19 September, 2018

Submitted By:

Name: Farid Uddin Ahmed

Class ID: 02

Name: Zarin Tasnim Sandhie

Class ID: 26



LIST OF THE DOCUMENTATIONS:

1. Author
2. Objective
3. Features
4. Configuration
5. Input/ Output Screenshots
6. Implementation & Code Snippet
7. Evaluation
8. Conclusion
9. References

AUTHORS

This report contains all the documents for the of Lab Assignment #1. The assignment was done by Farid Uddin Ahmed (Class ID 2) and Zarin Tasnim Sandhie (Class ID 26), both are graduate student majoring in Electrical Engineering Department at University of Missouri Kansas City (UMKC).

OBJECTIVE

In the first four weeks of Python/deep learning class, we got to work with the basic functions of python software. The definition and working principle of the following topics were taught in the class:

* Conditional statement
* Function
* Loop
* Set, List, Tuple
* Class
* Web parsing

This assignment includes all of the things mentioned above.

* The first problem deals with the use of “Conditional Statement”.
* The second problem deals with the use of “File handling” and “Conditional Statement”.
* For solving the third problem, “Set” “List” and “Function” were used.
* The fourth problem works on the basis of “Class” and “Inheritance”.
* And the last problem is solely based on “Web parsing” and the use of library “BeautifulSoup”. The five parts of these assignment are described below.

FEATURES

The features of all the problems are discussed below:

Problem 1:

Search in a string and find the first non-repeated characters in that string.

Problem 2:

Program a code such to remove everything in the File1 which is inside File2.

Problem 3:

Objective: Consider the following scenario. You have a list of students who are attending class "Python" and another list of students who are attending class "Web Application".Find the list of students who are attending “python” classes but not “Web Application”.

Problem 4:

Write a python program to create the following management systems. a. Hospital admission System (e.g. classes Patient, Doctor, Medical Admission Clerk, Book, Nurse,etc.). Your code should have: at least five classes, init constructor in all the classes, show inheritance, one super call, Use of self, Use of at least one private data member, Use of multiple Inheritance.

Problem 5:

Program a code which download a webpage contains a table using Request library, then parse the page using Beautifusoup library. You should save all the information of the table in a file.

CONFIGURATION

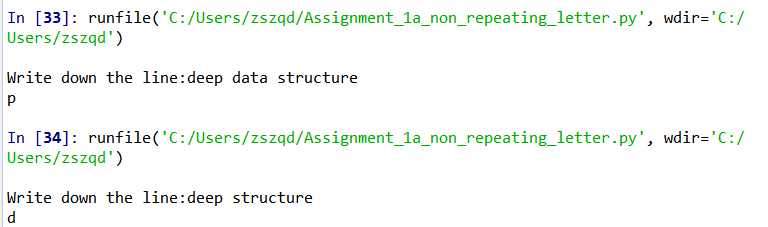
For executing the assignment, coding was done with Python software version 3.6. The simulation was in software: Anaconda (Spider).

INPUT/OUTPUT SCREENSHOTS

Problem 1:

* “deep data structure” is given as input.
* It finds the first non-repeating letter which is “p” and prints it.
* For the input “deep structure”, it gives the output “d”.

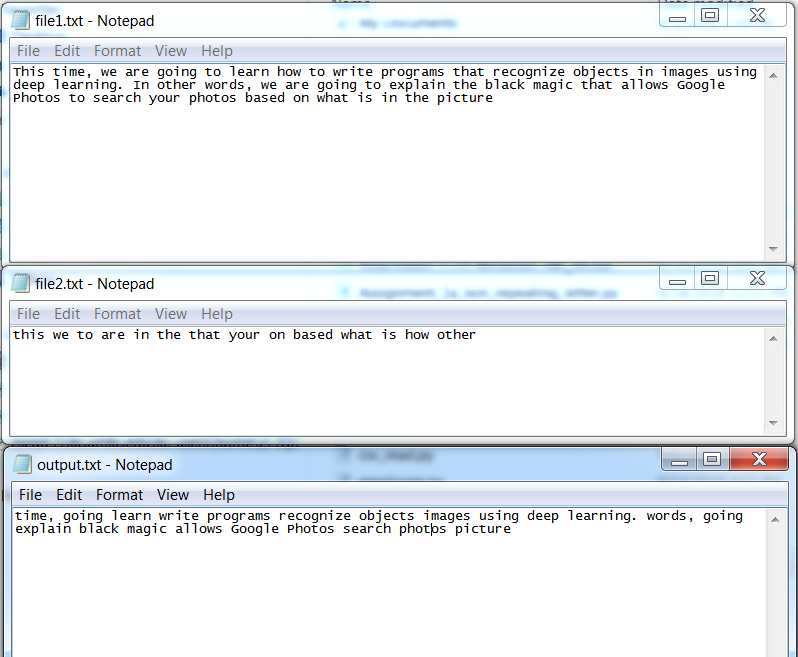
Output:



Problem 2:

* It reads two files and omits the words in the second file from the first file and prints the output in output file.
* The first input file is named as “file1.txt”.
* The second file is “file2.txt”.
* It omits the letters in “file2.txt” from “file1.txt” and stores in “output.txt”

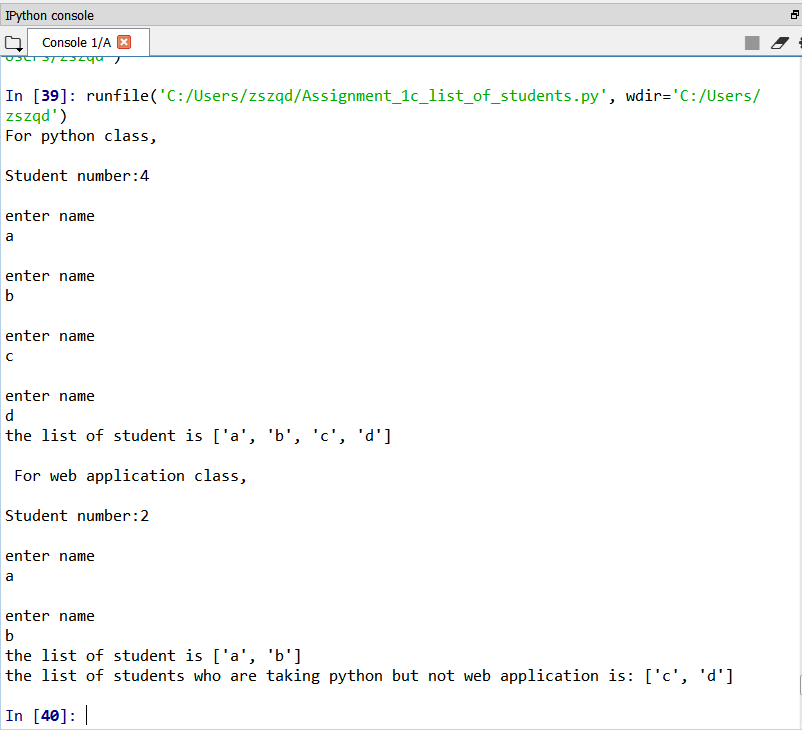
Output:



Problem 3:

* Two lists are taken as input which represents the students of two different classes.
* One class is for students attending the “Python” class (e.g. [‘a’, ‘b’, ‘c’, ‘d’]) and another class is for students attending the “Web application” class (e.g. [‘a’, ‘b’]).
* It gives out the list of students who took “Python” but not “Web application” (i.e. [‘c’, ‘d’]).

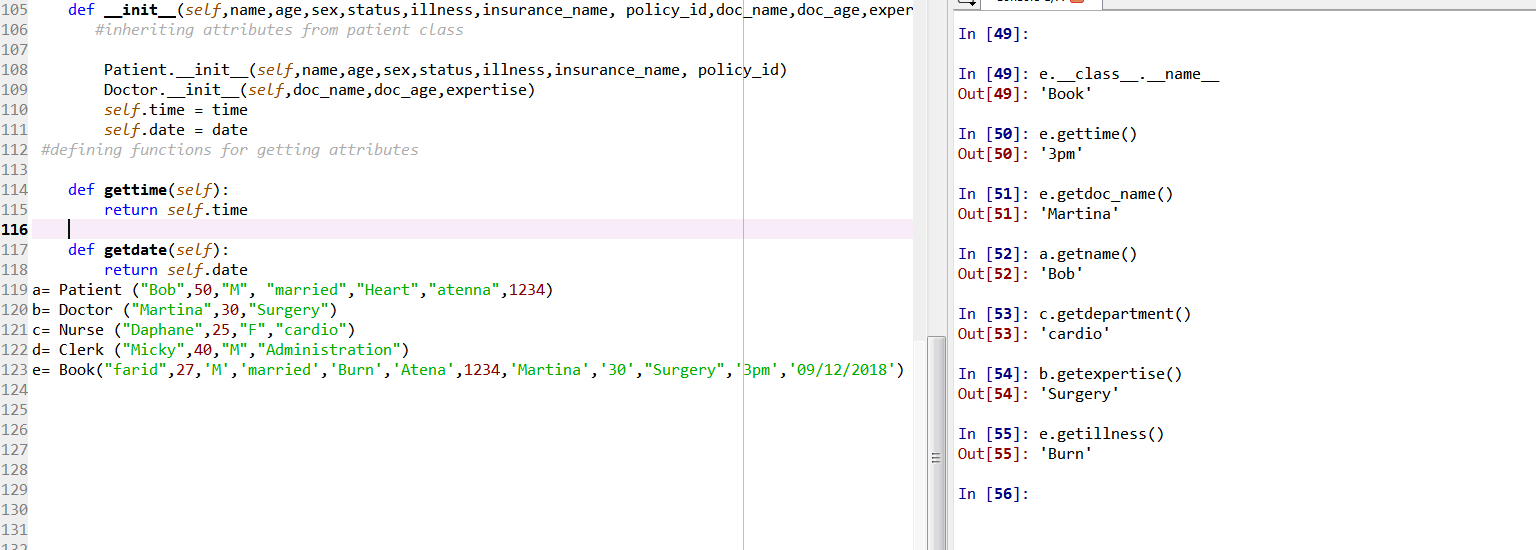
Output:



Problem 4:

* Different functions like “getname”, “getexpertise”, “gettime” etc. were called for different classes.
* To observe the name of the class for a particular element “e”, “e.\_\_class\_\_.\_\_name\_\_” was called.

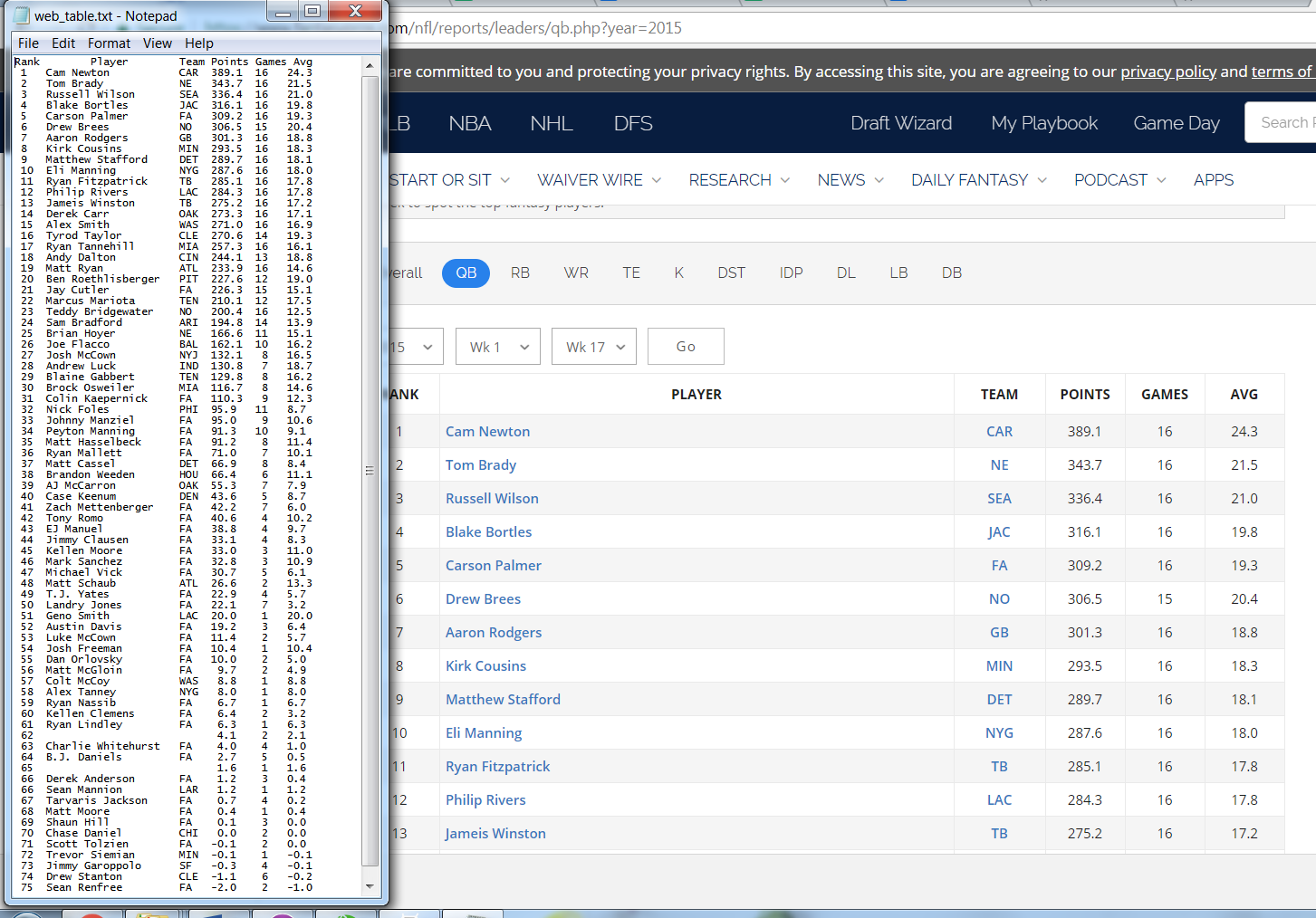
Output:



Problem 5:

* The link of the page containing the table is given.
* The output table shows the table extracted from the link.
* All the required elements of the table are shown.

Output:

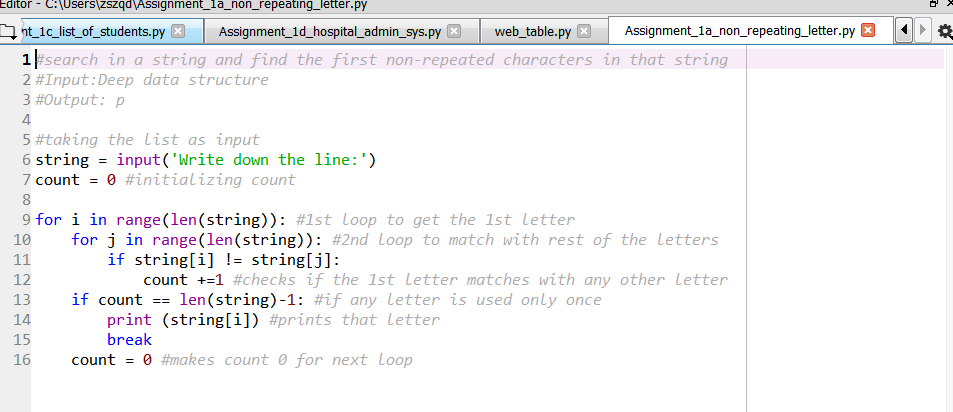


IMPLEMENTATION & CODE SNIPPET

Problem 1:

* The code take the string as input.
* Two loops are used in code.
* For the first loop, it take the letters of the sentence one by one.
* Then it matches with the rest of the letters in the sentences in the second loop.
* If it finds any match, it increases the count.
* Then it matches if the number of count is one.
* If it is one, it prints the letter.

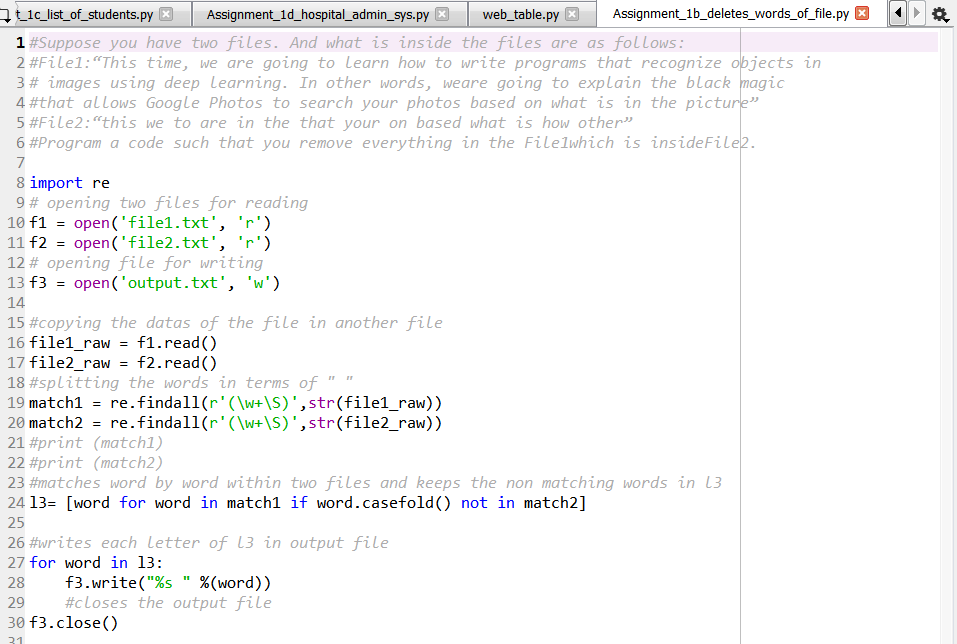
Code:



Problem 2:

* Here, two files File1 and File2 are opened for reading.
* And another file output is opened for writing.
* The words of File1 and File2 are seperated interms of spaces.
* Then the files are matched word by words.
* The match is case insensitive.
* The unmatched words are kept in a list.
* Then the words are printed in the output file one by one.

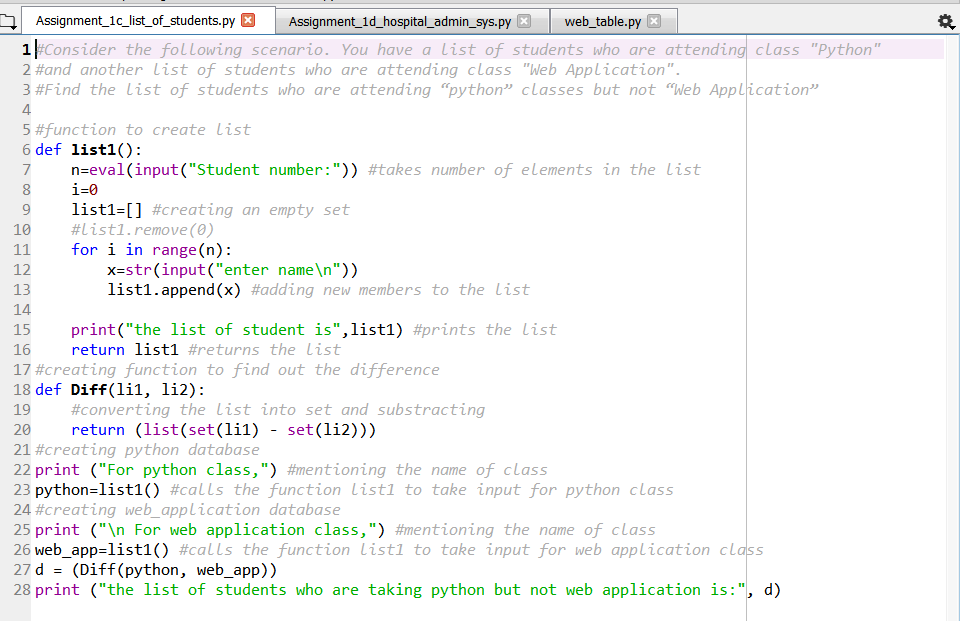
Code:



Problem 3:

* A function "list1" is declared which can take a list.
* Another function "Diff" is declared which takes two lists as inputs and converts the lists into set and gives the difference of the lists.
* The function "list1" is called twice to take the input of the lists of the students who are taking "Python" and who are taking "Web application".
* The "Diff" function is called to get the list of the students who is attending the class "Python" but not "Web application"

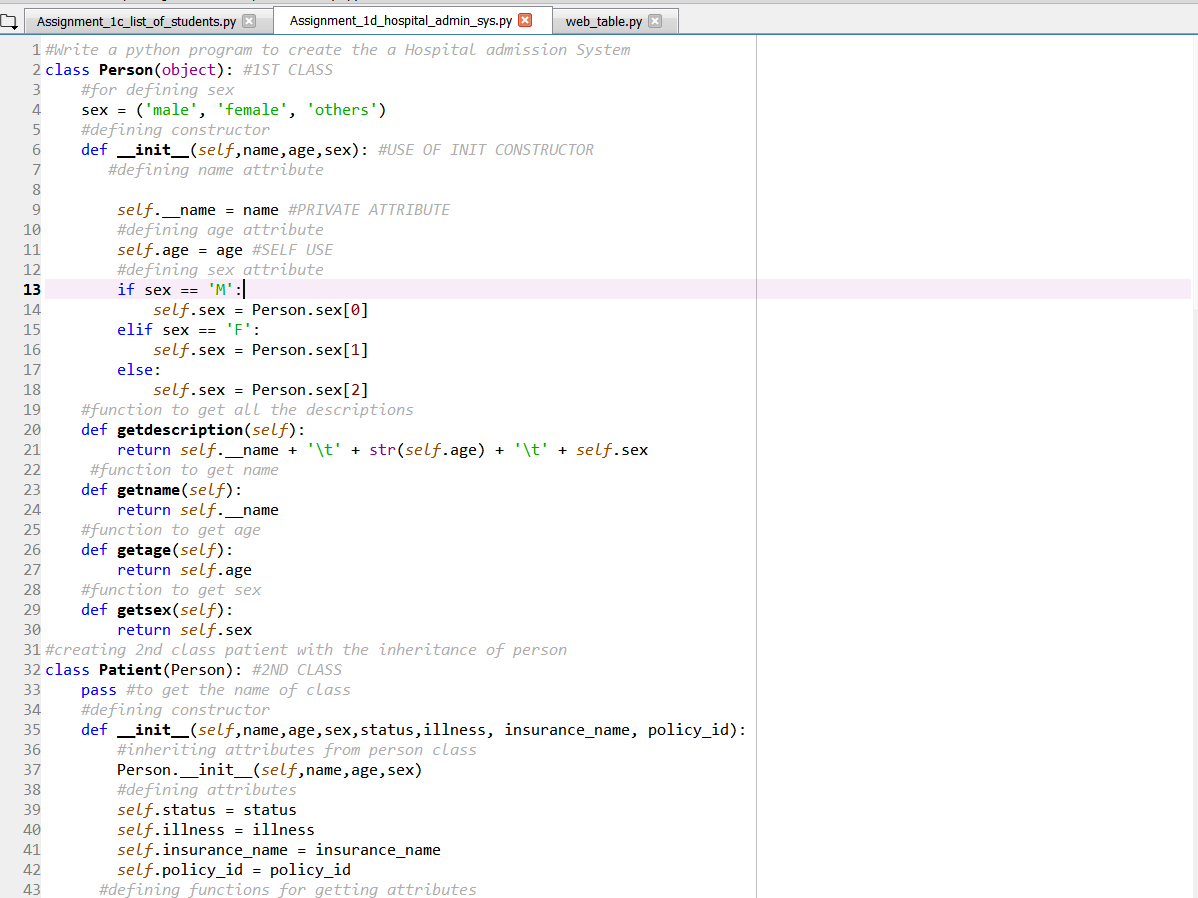
Code:



Problem 4:

* A Person class is defined which is to be used for inheritance by other classes. It has three attributes: name, age, sex.
* The class Patient is defined which inherits the properties of class Person. Also it has four new attributes: status, illness, insurance\_name, policy\_id.
* A Doctor class is declared which has the attributes doc\_name, doc\_age and expertise.
* A Nurse class is declared which inherits the properties of class Person. Also it has one new attribute: department.
* A Clerk class is declared which inherits the properties of class Person. And it has one new attribute: department.
* A Book class is declared which inherits the properties of class Patient and class Doctor. This is an example of multiple inheritance.
* “\_init\_constructor” is used in all the classes.
* “Super call” is used in the class Nurse.
* “Self” is used in every class.
* “Private data member” is used during defining name of person.

Code:



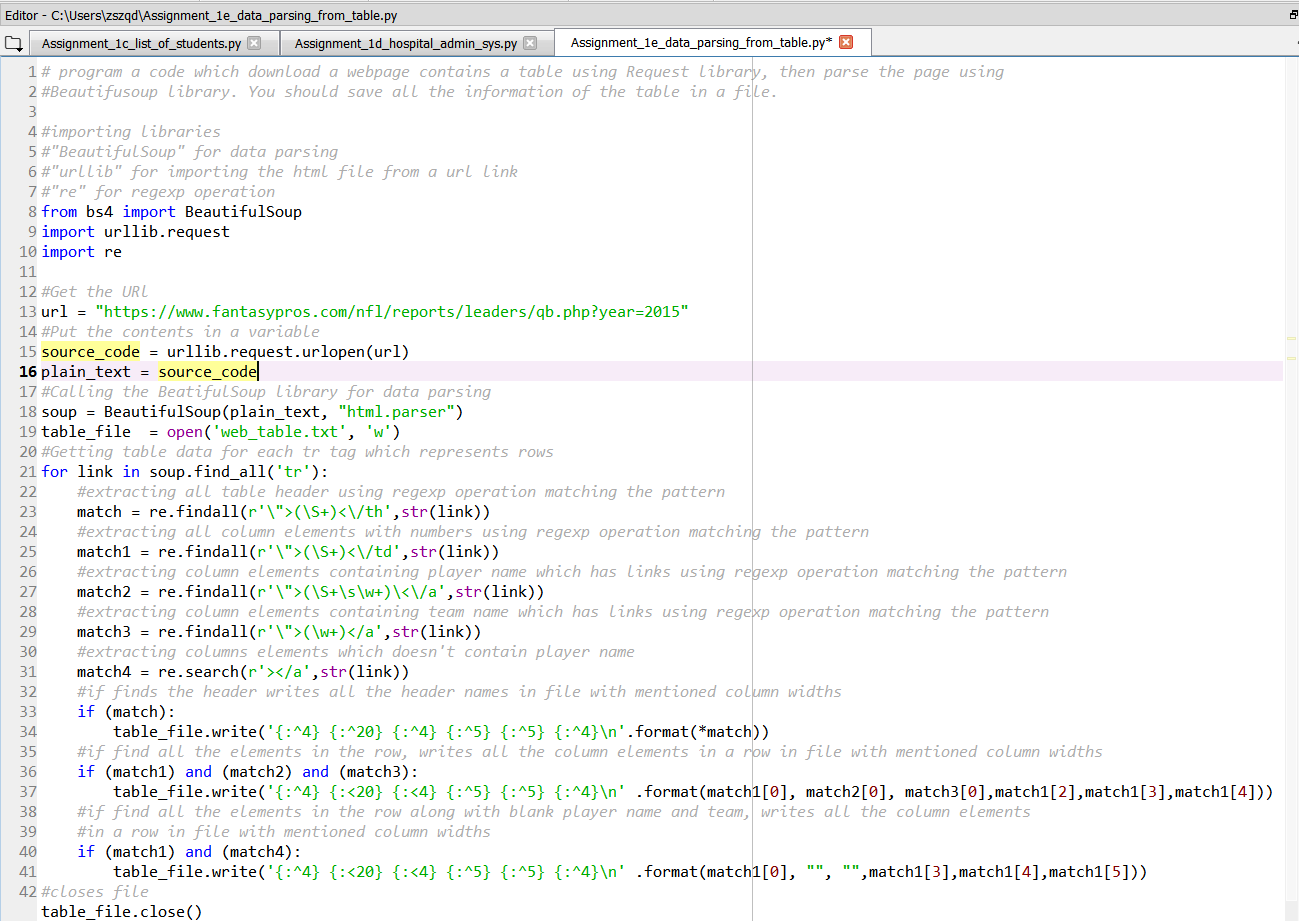




Problem 5:

* At first, the required libraries are imported. "BeautifulSoup" for data parsing, "urllib" for importing the html file from a url link,"re" for regexp operation.
* The contents of the url is taken into a variable.
* BeautifulSoup is used for parsing the elements.
* A file is opened for writing the column elements.
* Table data is obtained for each tr tag which represents rows.
* Table header is extracted using regexp operation and kept in "match".
* Column elements with numbers are extracted using regexp operation and kept in "match1".
* Column elements with player name which has links are extracted using regexp operation and kept in "match2".
* Column elements with team name which has links are extracted using regexp operation and kept in "match3".
* Column elements with blank player name and team name are extracted using regexp operation and kept in "match4".
* With if condition, header elements are placed in the file with appropriate column widths.
* With if condition, column elements of a row are placed in the file with appropriate column widths.
* With if condition, column elements of a row which has a blank player and team name are placed in the file with appropriate column widths.
* The output file is closed.

Code:



EVALUATION:

Problem 1:

The code shows the exact input output relation that was mentioned in the assignment objective. Different types of strings were used for the validation if the code and it was able to find the first non-repeating letter from all of the string types.

Problem 2:

The code generates the exact file which was supposed to be generated according to the objective. The matching operation is case insensitive as per instruction.

Problem 3:

The input output relationship matches the exact requirement mentioned in the assignment problem. The code is written to be used with any number of input students. Instead of using the student list inside the code, we have taken the list as an variable input which makes the code more easier to use.

Problem 4:

The code contains all the required specifications. For the verification of the output, we have showed only one element per class. But it can be modified with any number of elements per class.

Problem 5:

The output file shows that the table elements are parsed properly. The code can be used for any kinds of links with the similar table elements and it can be modified to be used for a wide range of links containing tables.

CONCLUSION:

All the required problems were solved successfully during this assignment. Through this assignment, we got to work with the basic commands and properties of “Python”. For this assignment, we used the software “Anaconda”. And all the required files and videos are documented in the Github link: <https://github.com/Sandhie177/CS5690-Python-deep-learning/tree/master/Assignment_1>.

REFERENCES:

1. <https://docs.python.org/3/library/re.html>
2. <https://docs.python.org/3/tutorial/classes.html>3.
3. <https://www.programiz.com/python-programming/inheritance>
4. <https://medium.freecodecamp.org/how-to-scrape-websites-with-python-and-beautifulsoup-5946935d93fe>