Team: 14
Professor: Yugyung Lee

Name: Sneha Mishra
Class ID: 19
Email: smccr@mail.umkc.edu
MyGitHub

Technical Partner:
Name: Raju Nekadi
Class ID: 20
Email: rn8mh@mail.umkc.edu
GitHub

Source code for this lab work can be found here

Objective

This Lab work is divided into 5 parts, to get familiar with python and understand the basic topics. Few of the topics covered in this lab are:

- Loop structures, Conditional statements, Functions, Tuples
- Complex datatypes, Dictionary, Sets, Python Functions, Web Scraping
- Classes, Object Oriented Concepts, Inheritance, instances, Scientific packages
- Data types, Operators, Conditional Statements

Features

Part 1:

Problem Statement:

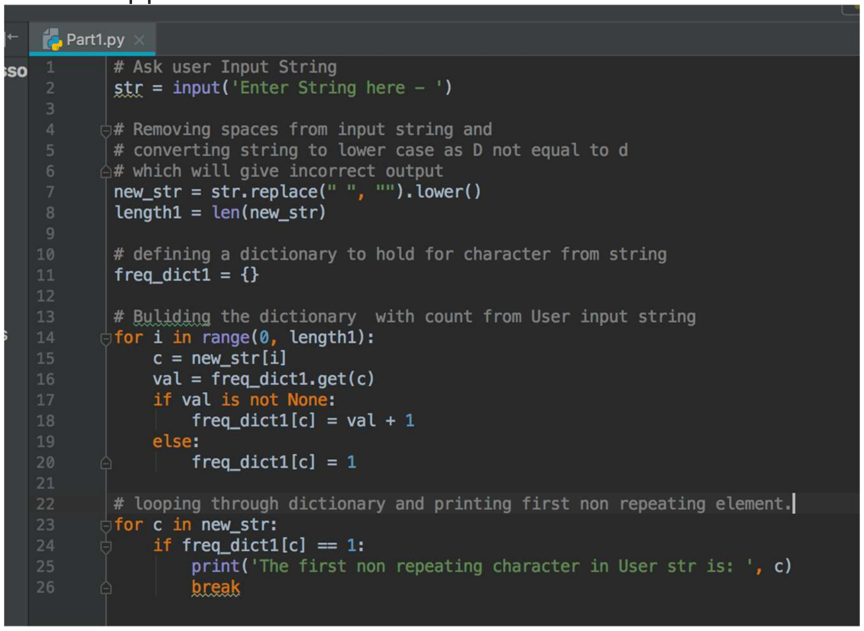
Search in a string and find the first non-repeated characters in that string.
Example-
Input: Deep data structure.
Output: p

(hint: if there is space in the string you need to consider the whole as one string. In the above example Deepdatastructure).

Solution:

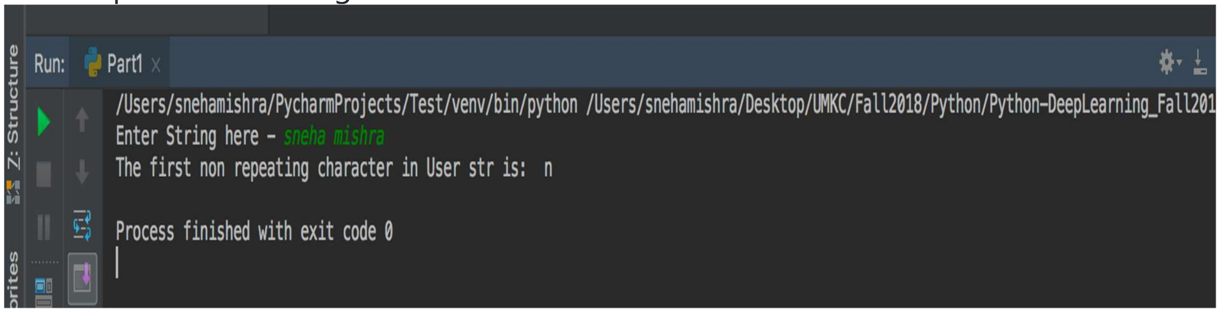
1. Ask user to input a string of choice.
2. Convert the string to lower case (can go for upper case if you want) to make it case in-sensitive.
3. Form a dictionary of each character in the string with its count of occurrences.
4. Loop through this list to get the first character which does not repeat.

Code snippet is attached below:



```python
# Ask user Input String
str = input('Enter String here - ')

# Removing spaces from input string and
# converting string to lower case as D not equal to d
# which will give incorrect output
new_str = str.replace(" ", "").lower()
length1 = len(new_str)

# defining a dictionary to hold for character from string
freq_dict1 = {}

# Buliding the dictionary  with count from User input string
for i in range(0, length1):
    c = new_str[i]
    val = freq_dict1.get(c)
    if val is not None:
        freq_dict1[c] = val + 1
    else:
        freq_dict1[c] = 1

# looping through dictionary and printing first non repeating element.
for c in new_str:
    if freq_dict1[c] == 1:
        print('The first non repeating character in User str is: ', c)
        break
```

The output after running the code is below:



```
/Users/snehamishra/PycharmProjects/Test/venv/bin/python /Users/snehamishra/Desktop/UMKC/Fall2018/Python/Python-DeepLearning_Fall201
Enter String here - sneha mishra
The first non repeating character in User str is:  n

Process finished with exit code 0
```

Part 2:

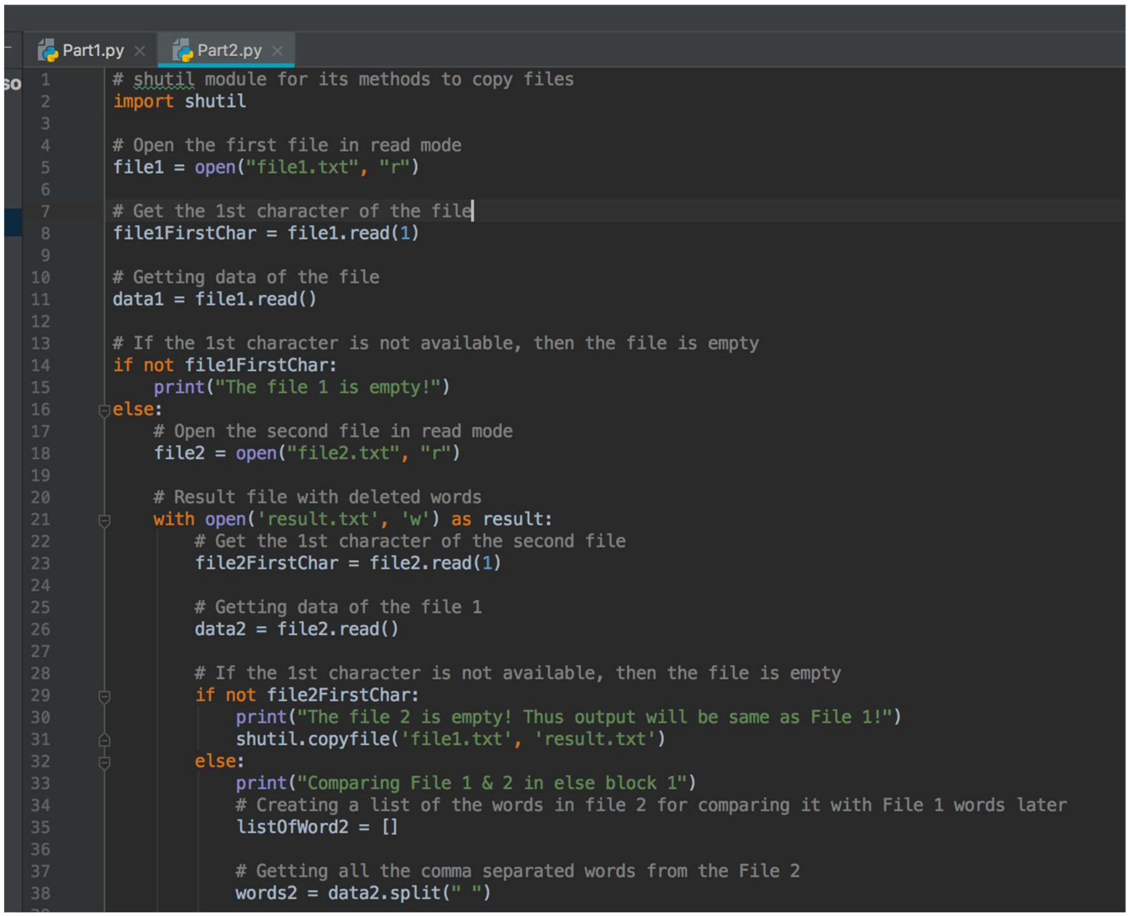
Problem Statement:

Suppose you have two files:
File1:“This time, we are going to learn how to write programs that recognize objects in images using deep learning. In other words, we are going to explain the black magic that allows Google Photos to search your photos based on what is in the picture”
File2:“this we to are in the that your on based what is how other”
Program a code such that you remove everything in the File1 which is inside File2.
The output of File1 will be: “time, going learn write programs recognize objects images using deep learning. words, going explain black magic allows Google Photos search photos picture".

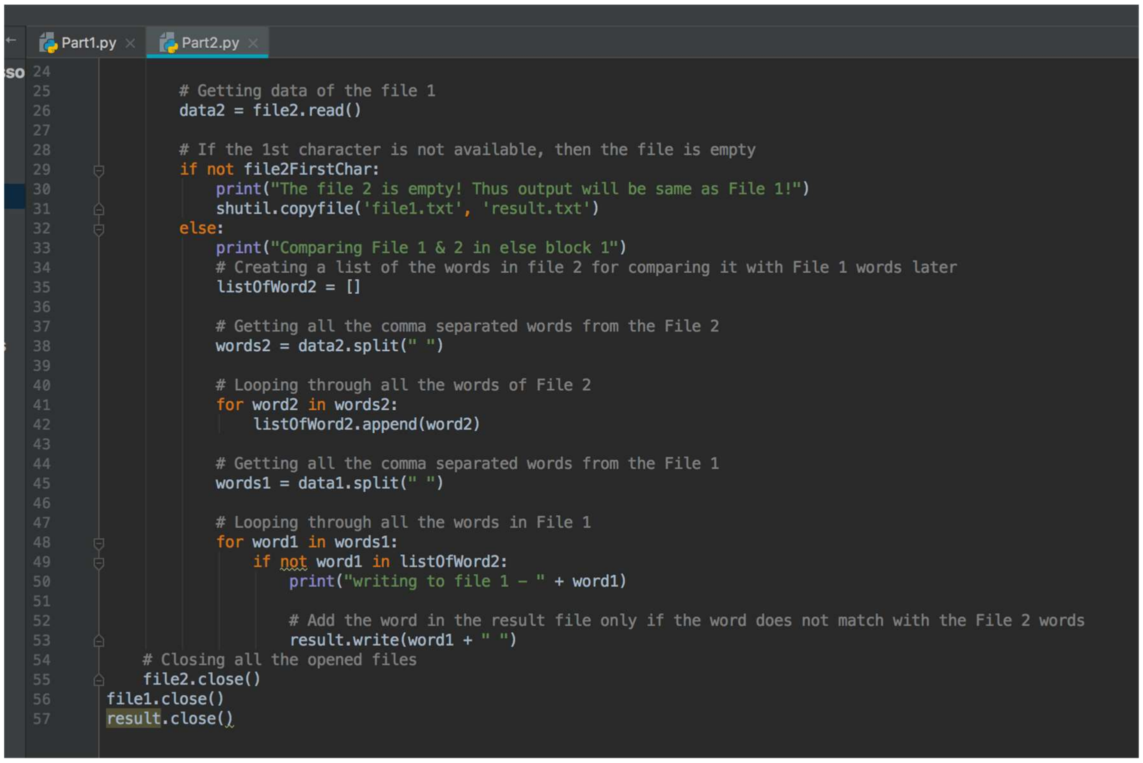
Solution:

1. Open both the input files in read mode (since these files are never updated, just read).
2. Check if the opened files are empty or not, if empty then display appropriate user messages.
3. When file not empty, Open a result file in write mode to update the File 1 contents.
4. Create a list of words from second file, which needs to be discarded from file 1.
5. Check if the word in file 1 is not in the above mentioned list of words, then only write the words in result file.

Code snippet is attached below:



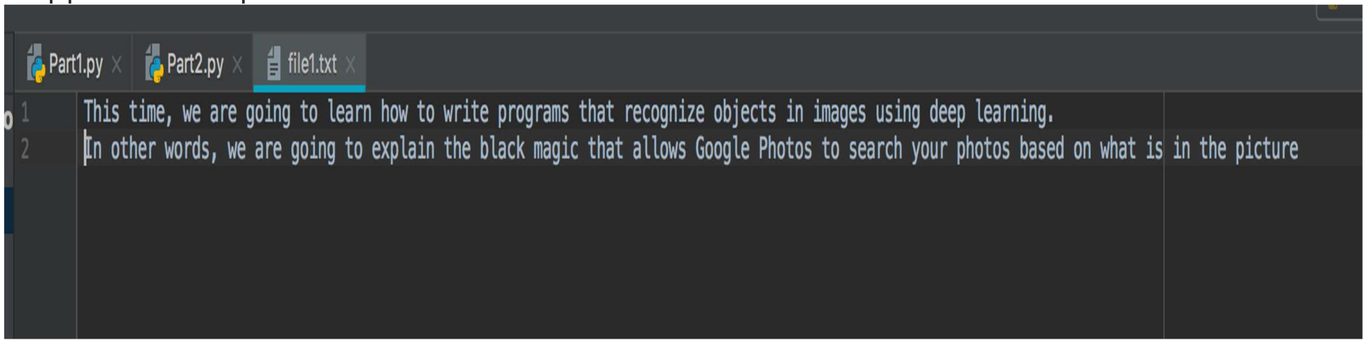
```python
# shutil module for its methods to copy files
import shutil

# Open the first file in read mode
file1 = open("file1.txt", "r")

# Get the 1st character of the file
file1FirstChar = file1.read(1)

# Getting data of the file
data1 = file1.read()

# If the 1st character is not available, then the file is empty
if not file1FirstChar:
    print("The file 1 is empty!")
else:
    # Open the second file in read mode
    file2 = open("file2.txt", "r")

    # Result file with deleted words
    with open('result.txt', 'w') as result:
        # Get the 1st character of the second file
        file2FirstChar = file2.read(1)

        # Getting data of the file 1
        data2 = file2.read()

        # If the 1st character is not available, then the file is empty
        if not file2FirstChar:
            print("The file 2 is empty! Thus output will be same as File 1!")
            shutil.copyfile('file1.txt', 'result.txt')
        else:
            print("Comparing File 1 & 2 in else block 1")
            # Creating a list of the words in file 2 for comparing it with File 1 words later
            listOfWord2 = []

            # Getting all the comma separated words from the File 2
            words2 = data2.split(" ")
```



Part1.py Part2.py

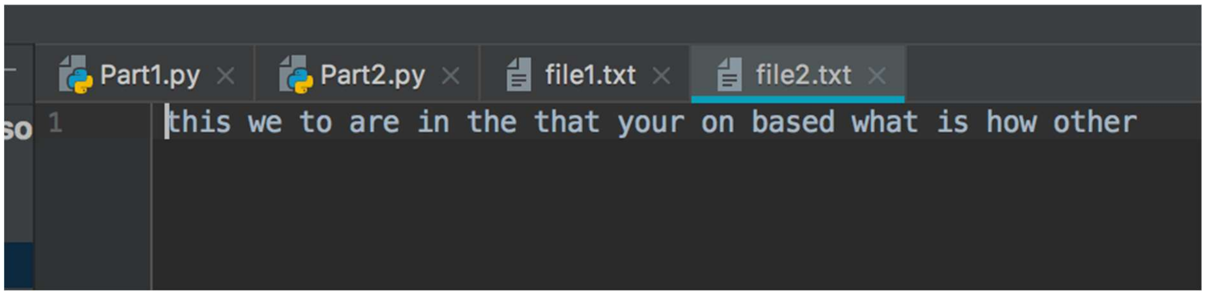
```python
        # Getting data of the file 1
        data2 = file2.read()

        # If the 1st character is not available, then the file is empty
        if not file2FirstChar:
            print("The file 2 is empty! Thus output will be same as File 1!")
            shutil.copyfile('file1.txt', 'result.txt')
        else:
            print("Comparing File 1 & 2 in else block 1")
            # Creating a list of the words in file 2 for comparing it with File 1 words later
            listOfWord2 = []

            # Getting all the comma separated words from the File 2
            words2 = data2.split(" ")

            # Looping through all the words of File 2
            for word2 in words2:
                listOfWord2.append(word2)

            # Getting all the comma separated words from the File 1
            words1 = data1.split(" ")

            # Looping through all the words in File 1
            for word1 in words1:
                if not word1 in listOfWord2:
                    print("writing to file 1 - " + word1)

                    # Add the word in the result file only if the word does not match with the File 2 words
                    result.write(word1 + " ")
    # Closing all the opened files
    file2.close()
file1.close()
result.close()
```

Snippet of the Input file to the code:



Part1.py Part2.py file1.txt

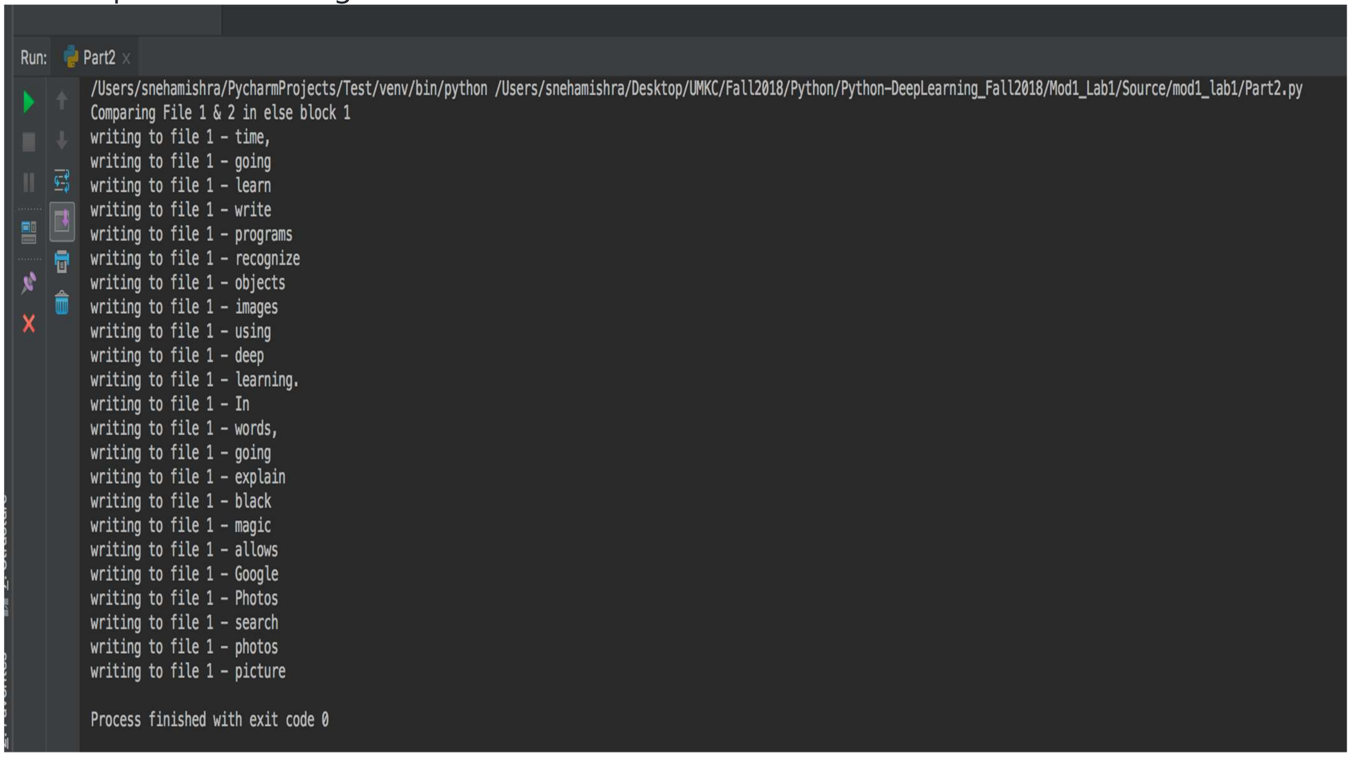
```
This time, we are going to learn how to write programs that recognize objects in images using deep learning.
In other words, we are going to explain the black magic that allows Google Photos to search your photos based on what is in the picture
```



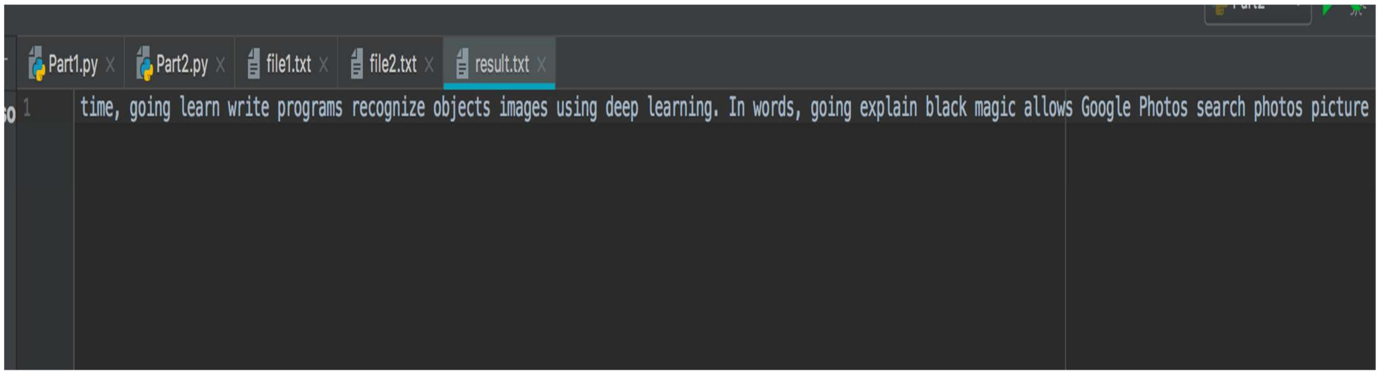
Part1.py Part2.py file1.txt file2.txt

```
this we to are in the that your on based what is how other
```

The output after running the code is below:

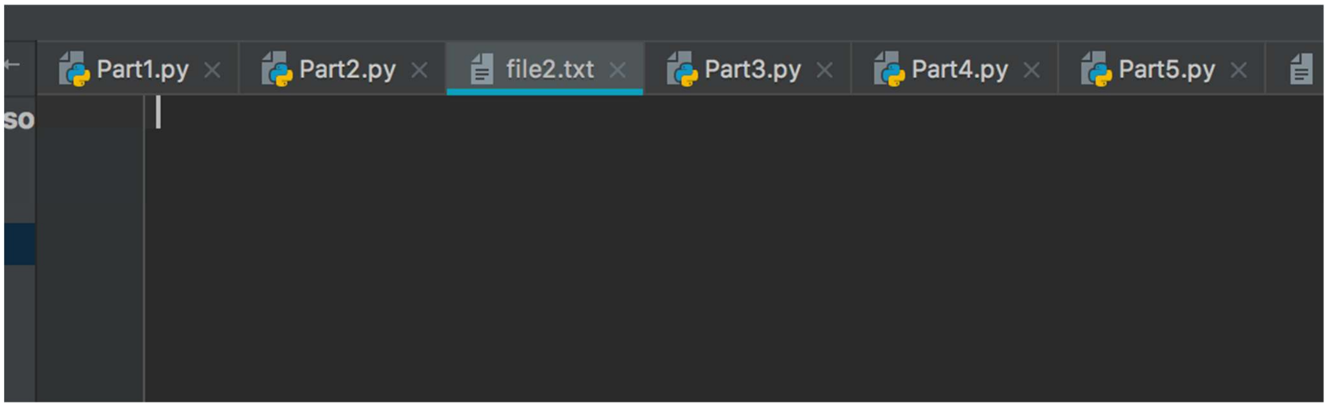


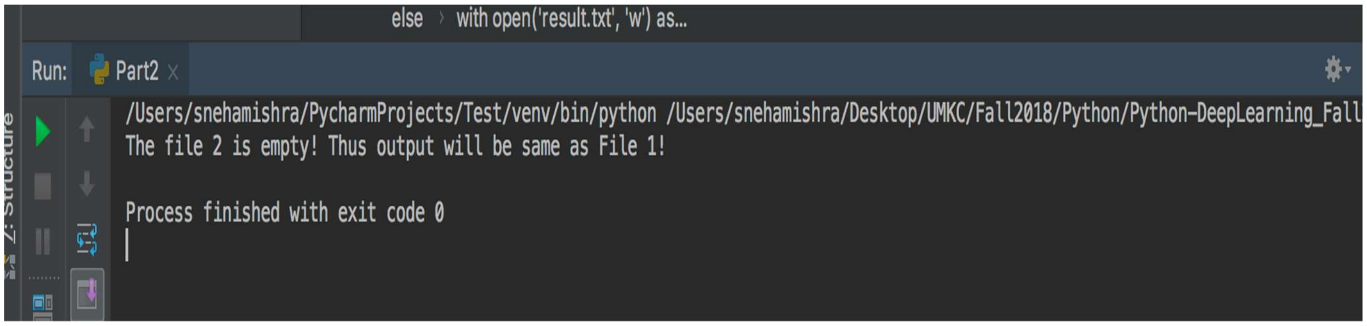
```
/Users/snehamishra/PycharmProjects/Test/venv/bin/python /Users/snehamishra/Desktop/UMKC/Fall2018/Python/Python-DeepLearning_Fall2018/Mod1_Lab1/Source/mod1_lab1/Part2.py
Comparing File 1 & 2 in else block 1
writing to file 1 - time,
writing to file 1 - going
writing to file 1 - learn
writing to file 1 - write
writing to file 1 - programs
writing to file 1 - recognize
writing to file 1 - objects
writing to file 1 - images
writing to file 1 - using
writing to file 1 - deep
writing to file 1 - learning.
writing to file 1 - In
writing to file 1 - words,
writing to file 1 - going
writing to file 1 - explain
writing to file 1 - black
writing to file 1 - magic
writing to file 1 - allows
writing to file 1 - Google
writing to file 1 - Photos
writing to file 1 - search
writing to file 1 - photos
writing to file 1 - picture

Process finished with exit code 0
```



```
time, going learn write programs recognize objects images using deep learning. In words, going explain black magic allows Google Photos search photos picture
```

When input file is empty, output will be:





```
/Users/snehamishra/PycharmProjects/Test/venv/bin/python /Users/snehamishra/Desktop/UMKC/Fall2018/Python/Python-DeepLearning_Fall
The file 2 is empty! Thus output will be same as File 1!

Process finished with exit code 0
```

Part 3:

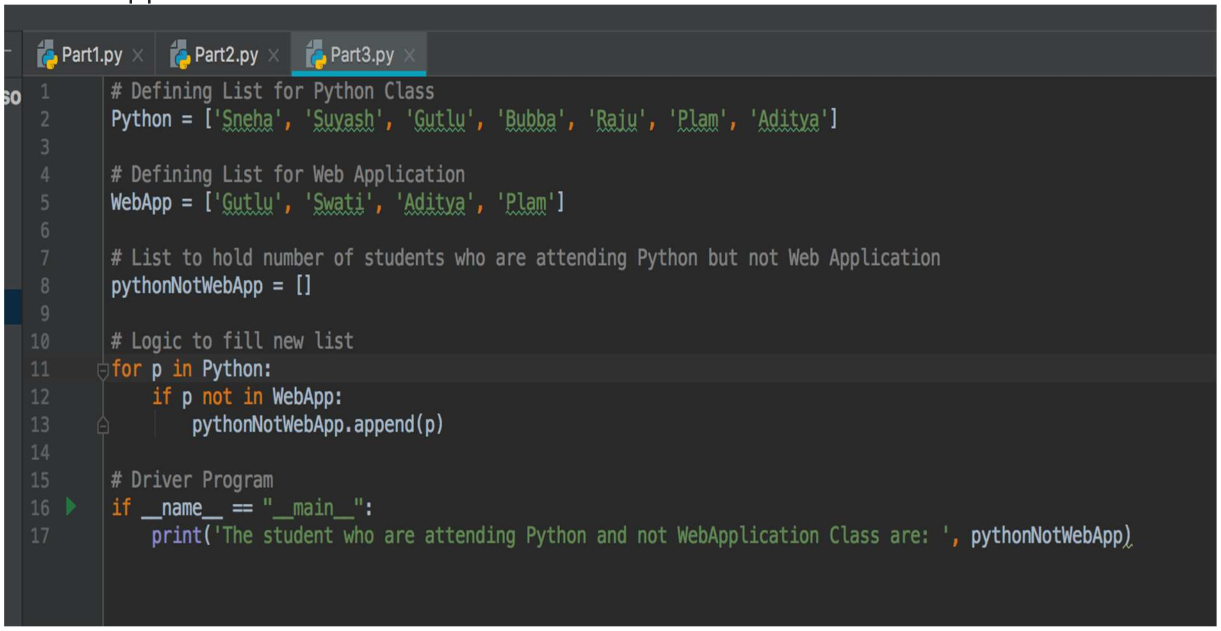
Problem Statement:

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”.

Solution:

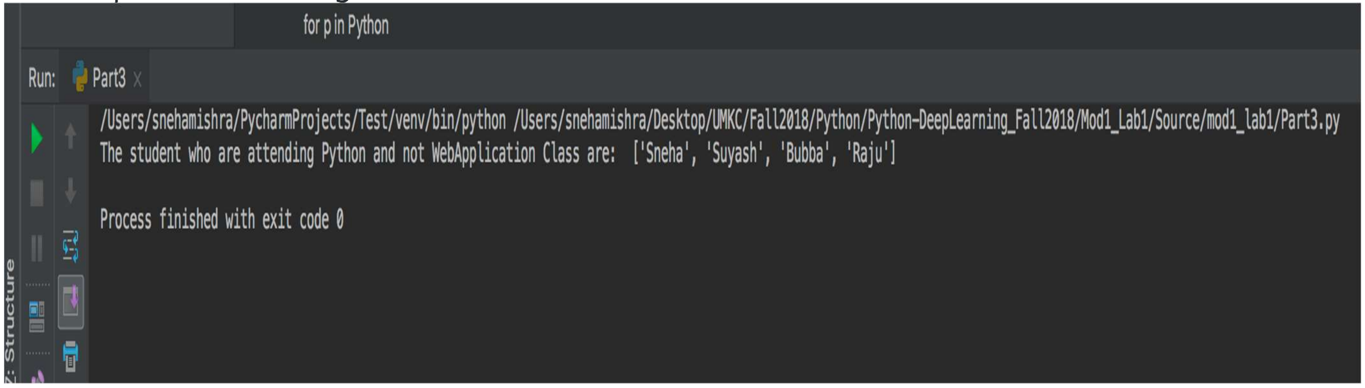
1. Create two lists of students for the two courses mentioned in the above problem.
2. Create a third list which contains the list of students enrolled in first course but not in the 2nd course.
3. Run the loop for the students in list one, and put the names of only those students who are not in the second list.
4. Print the third list.

Code snippet is attached below:



```python
# Defining List for Python Class
Python = ['Sneha', 'Suyash', 'Gutlu', 'Bubba', 'Raju', 'Plam', 'Aditya']

# Defining List for Web Application
WebApp = ['Gutlu', 'Swati', 'Aditya', 'Plam']

# List to hold number of students who are attending Python but not Web Application
pythonNotWebApp = []

# Logic to fill new list
for p in Python:
    if p not in WebApp:
        pythonNotWebApp.append(p)

# Driver Program
if __name__ == "__main__":
    print('The student who are attending Python and not WebApplication Class are: ', pythonNotWebApp)
```

The output after running the code is below:



```
/Users/snehamishra/PycharmProjects/Test/venv/bin/python /Users/snehamishra/Desktop/UMKC/Fall2018/Python/Python-DeepLearning_Fall2018/Mod1_Lab1/Source/mod1_lab1/Part3.py
The student who are attending Python and not WebApplication Class are:  ['Sneha', 'Suyash', 'Bubba', 'Raju']

Process finished with exit code 0
```

Part 4:

Problem Statement:

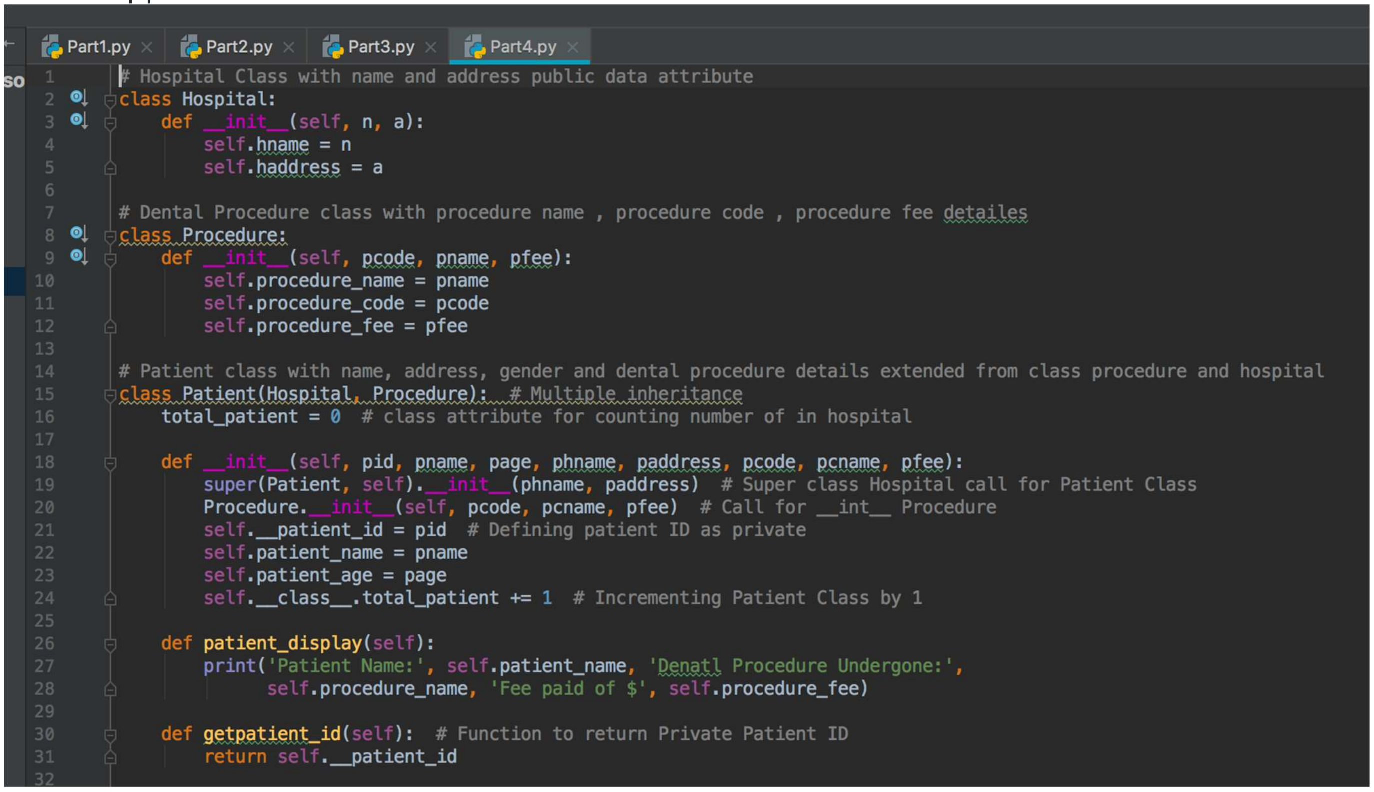
Write a python program to create a Hospital admission System (e.g. classes Patient, Doctor, Medical Admission Clerk, Book, Nurse, etc.)
Prerequisites:
a. Your code should have at least five classes
b. Your code should have *init* constructor in all the classes
c. Your code should show inheritance at least once
d. Your code should have one super call
e. Use of self is required

f. Use at least one private data member in your code
g. Use multiple Inheritance at least once
h. Create instances of all classes and show the relationship between them
Comment your code appropriately to point out where all these things are present

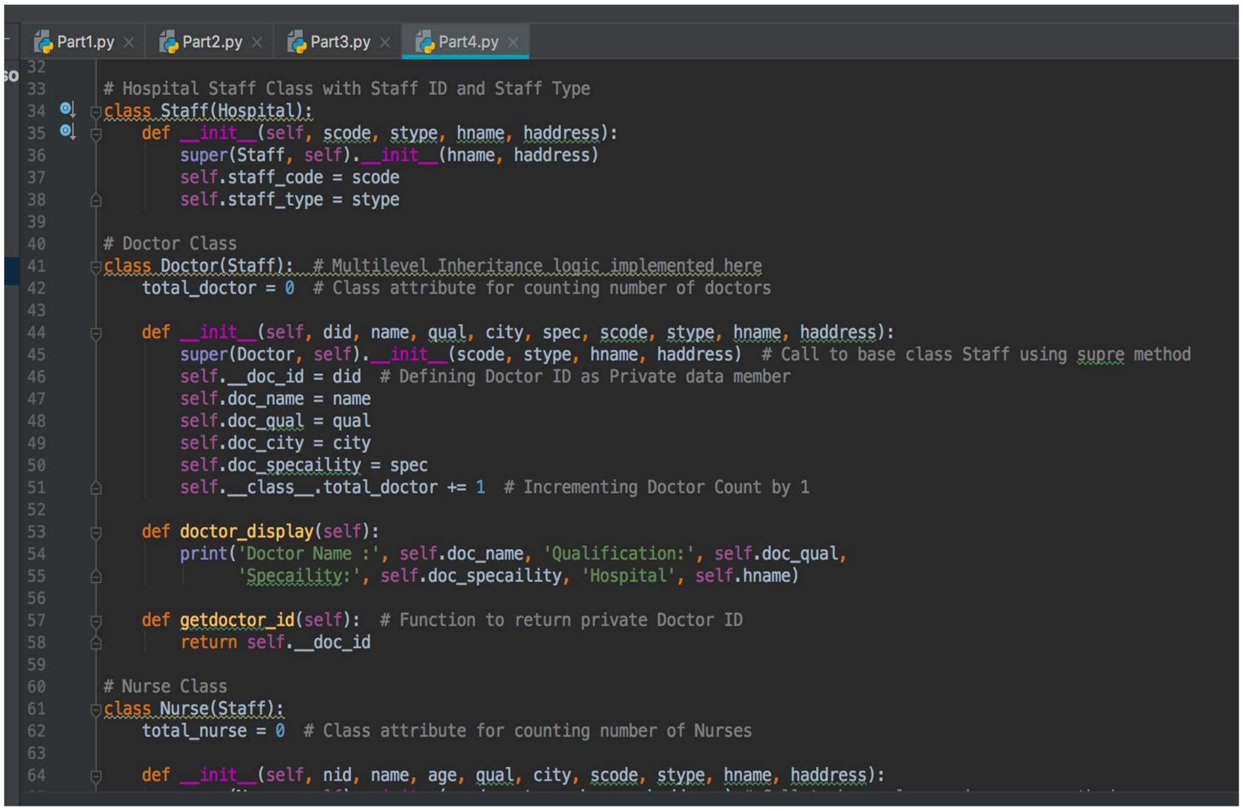
Solution:

1. Create Classes called Hospital, Procedure, Patient, Staff, Nurse and Doctor.
2. Define the attributes and getter/setter for each class as required.
3. For multiple inheritance, class Patient inherits properties from two classes Hospital and Procedure.
4. Class Staff inherits from class Hospital, while class Doctor and Nurse inherit from class Staff.
5. Create objects of the classes and print them out.

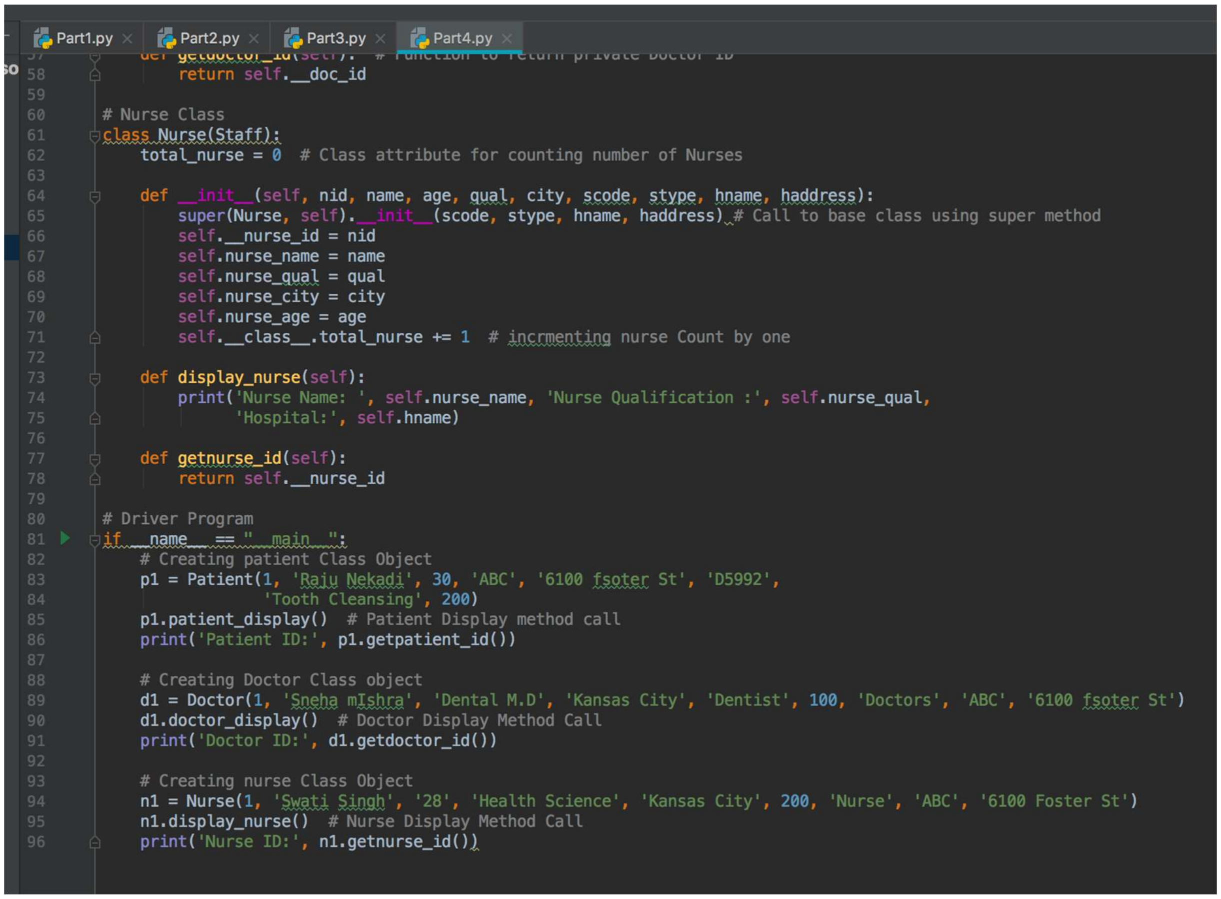
Code snippet is attached below:



```python
# Hospital Class with name and address public data attribute
class Hospital:
    def __init__(self, n, a):
        self.hname = n
        self.haddress = a

# Dental Procedure class with procedure name , procedure code , procedure fee detailes
class Procedure:
    def __init__(self, pcode, pname, pfee):
        self.procedure_name = pname
        self.procedure_code = pcode
        self.procedure_fee = pfee

# Patient class with name, address, gender and dental procedure details extended from class procedure and hospital
class Patient(Hospital, Procedure):  # Multiple inheritance
    total_patient = 0  # class attribute for counting number of in hospital

    def __init__(self, pid, pname, page, phname, paddress, pcode, pcname, pfee):
        super(Patient, self).__init__(phname, paddress)  # Super class Hospital call for Patient Class
        Procedure.__init__(self, pcode, pcname, pfee)  # Call for __int__ Procedure
        self.__patient_id = pid  # Defining patient ID as private
        self.patient_name = pname
        self.patient_age = page
        self.__class__.total_patient += 1  # Incrementing Patient Class by 1

    def patient_display(self):
        print('Patient Name:', self.patient_name, 'Denatl Procedure Undergone:',
              self.procedure_name, 'Fee paid of $', self.procedure_fee)

    def getpatient_id(self):  # Function to return Private Patient ID
        return self.__patient_id
```

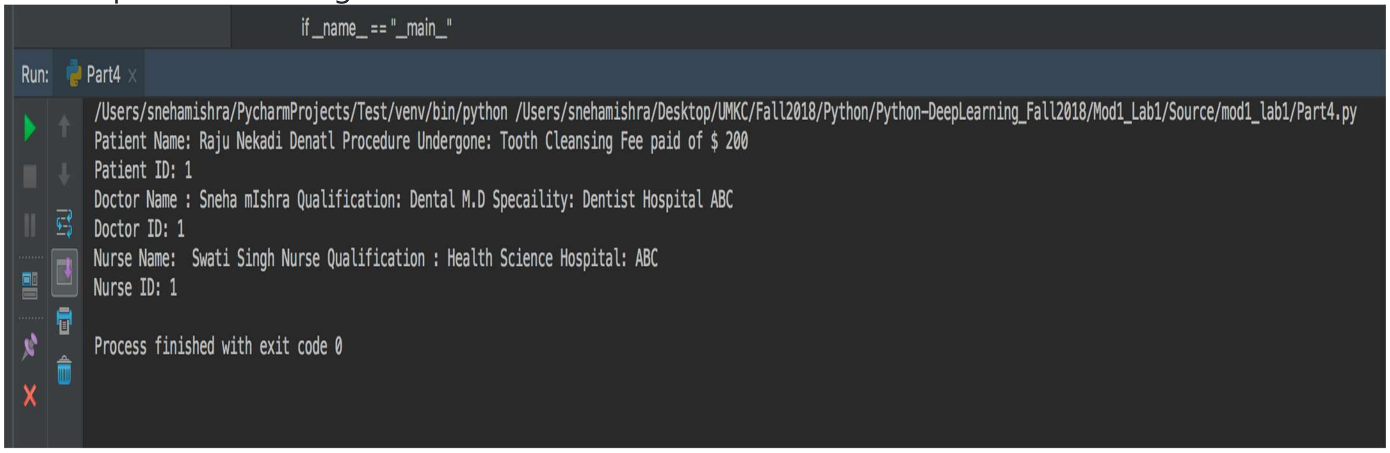


```python

# Hospital Staff Class with Staff ID and Staff Type
class Staff(Hospital):
    def __init__(self, scode, stype, hname, haddress):
        super(Staff, self).__init__(hname, haddress)
        self.staff_code = scode
        self.staff_type = stype

# Doctor Class
class Doctor(Staff):  # Multilevel Inheritance logic implemented here
    total_doctor = 0  # Class attribute for counting number of doctors

    def __init__(self, did, name, qual, city, spec, scode, stype, hname, haddress):
        super(Doctor, self).__init__(scode, stype, hname, haddress)  # Call to base class Staff using supre method
        self.__doc_id = did  # Defining Doctor ID as Private data member
        self.doc_name = name
        self.doc_qual = qual
        self.doc_city = city
        self.doc_specaility = spec
        self.__class__.total_doctor += 1  # Incrementing Doctor Count by 1

    def doctor_display(self):
        print('Doctor Name :', self.doc_name, 'Qualification:', self.doc_qual,
              'Specaility:', self.doc_specaility, 'Hospital', self.hname)

    def getdoctor_id(self):  # Function to return private Doctor ID
        return self.__doc_id

# Nurse Class
class Nurse(Staff):
    total_nurse = 0  # Class attribute for counting number of Nurses

    def __init__(self, nid, name, age, qual, city, scode, stype, hname, haddress):
```



```python
        return self.__doc_id

# Nurse Class
class Nurse(Staff):
    total_nurse = 0  # Class attribute for counting number of Nurses

    def __init__(self, nid, name, age, qual, city, scode, stype, hname, haddress):
        super(Nurse, self).__init__(scode, stype, hname, haddress) # Call to base class using super method
        self.__nurse_id = nid
        self.nurse_name = name
        self.nurse_qual = qual
        self.nurse_city = city
        self.nurse_age = age
        self.__class__.total_nurse += 1  # incrmenting nurse Count by one

    def display_nurse(self):
        print('Nurse Name: ', self.nurse_name, 'Nurse Qualification :', self.nurse_qual,
              'Hospital:', self.hname)

    def getnurse_id(self):
        return self.__nurse_id

# Driver Program
if __name__ == "__main__":
    # Creating patient Class Object
    p1 = Patient(1, 'Raju Nekadi', 30, 'ABC', '6100 fsoter St', 'D5992',
                 'Tooth Cleansing', 200)
    p1.patient_display()  # Patient Display method call
    print('Patient ID:', p1.getpatient_id())

    # Creating Doctor Class object
    d1 = Doctor(1, 'Sneha mIshra', 'Dental M.D', 'Kansas City', 'Dentist', 100, 'Doctors', 'ABC', '6100 fsoter St')
    d1.doctor_display()  # Doctor Display Method Call
    print('Doctor ID:', d1.getdoctor_id())

    # Creating nurse Class Object
    n1 = Nurse(1, 'Swati Singh', '28', 'Health Science', 'Kansas City', 200, 'Nurse', 'ABC', '6100 Foster St')
    n1.display_nurse()  # Nurse Display Method Call
    print('Nurse ID:', n1.getnurse_id())
```

The output after running the code is below:



```
if _name_ == "_main_"
```

Run: Part4

```
/Users/snehamishra/PycharmProjects/Test/venv/bin/python /Users/snehamishra/Desktop/UMKC/Fall2018/Python/Python-DeepLearning_Fall2018/Mod1_Lab1/Source/mod1_lab1/Part4.py
Patient Name: Raju Nekadi Denatl Procedure Undergone: Tooth Cleansing Fee paid of $ 200
Patient ID: 1
Doctor Name : Sneha mIshra Qualification: Dental M.D Specaility: Dentist Hospital ABC
Doctor ID: 1
Nurse Name:  Swati Singh Nurse Qualification : Health Science Hospital: ABC
Nurse ID: 1

Process finished with exit code 0
```

Part 5:

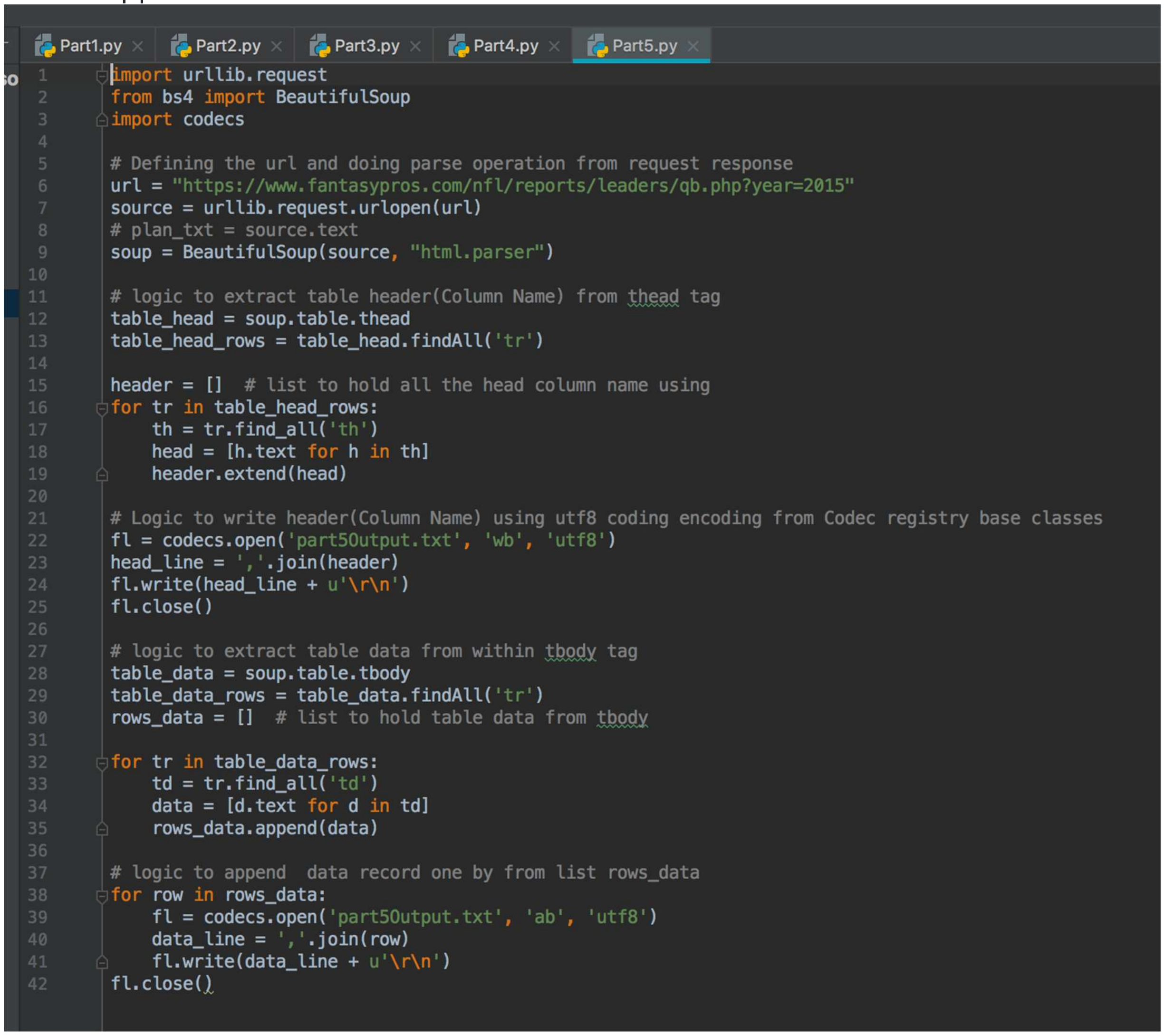
Problem Statement:

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.
Sample input: https://www.fantasypros.com/nfl/reports/leaders/qb.php?year=2015
Sample output: Save the table in this link into a file

Solution:

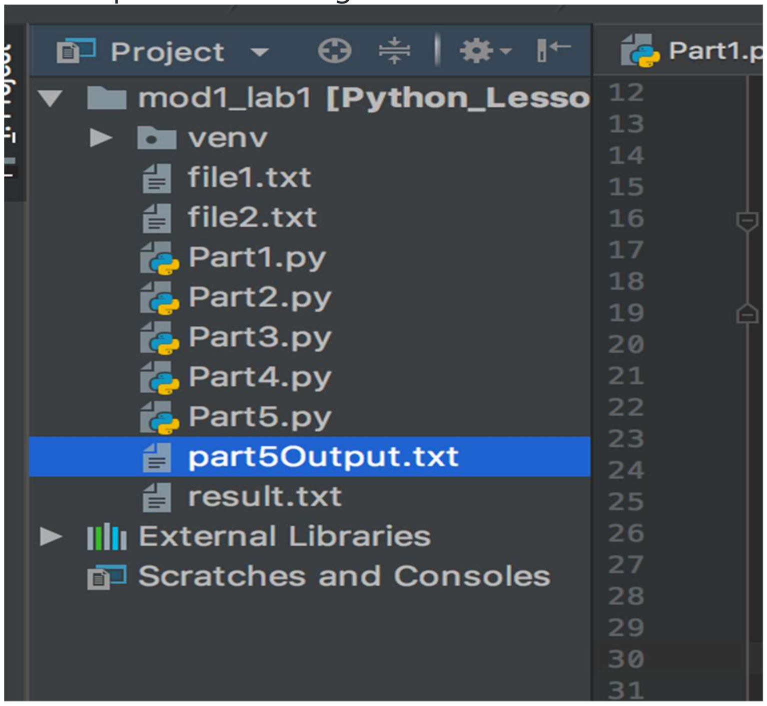
1. Get the url and fetch the contents of that web page.
2. The html is parsed using the BeautifulSoup libraries.
3. Open another text file as output to save the contents parsed from this web page.
4. Extract the data from the given url using the inbuilt functions and write them to the output text file.

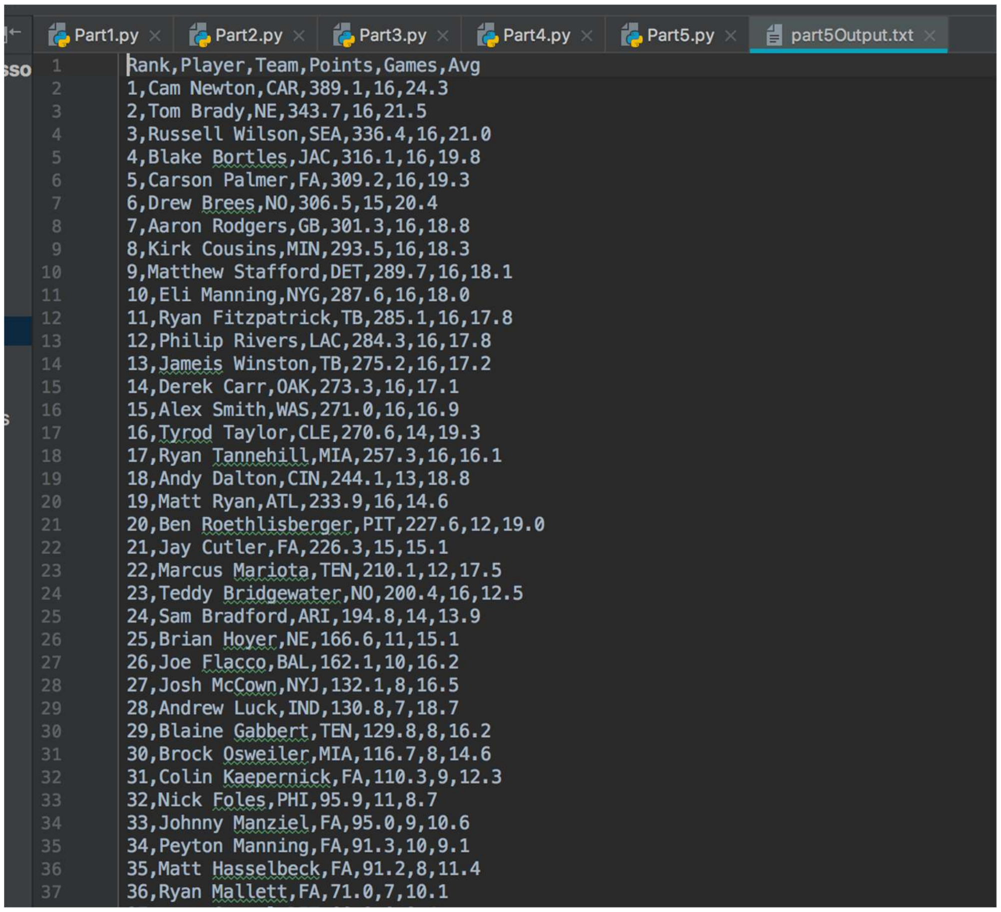
Code snippet is attached below:



```python
import urllib.request
from bs4 import BeautifulSoup
import codecs

# Defining the url and doing parse operation from request response
url = "https://www.fantasypros.com/nfl/reports/leaders/qb.php?year=2015"
source = urllib.request.urlopen(url)
# plan_txt = source.text
soup = BeautifulSoup(source, "html.parser")

# logic to extract table header(Column Name) from thead tag
table_head = soup.table.thead
table_head_rows = table_head.findAll('tr')

header = []  # list to hold all the head column name using
for tr in table_head_rows:
    th = tr.find_all('th')
    head = [h.text for h in th]
    header.extend(head)

# Logic to write header(Column Name) using utf8 coding encoding from Codec registry base classes
fl = codecs.open('part5Output.txt', 'wb', 'utf8')
head_line = ','.join(header)
fl.write(head_line + u'\r\n')
fl.close()

# logic to extract table data from within tbody tag
table_data = soup.table.tbody
table_data_rows = table_data.findAll('tr')
rows_data = []  # list to hold table data from tbody

for tr in table_data_rows:
    td = tr.find_all('td')
    data = [d.text for d in td]
    rows_data.append(data)

# logic to append  data record one by from list rows_data
for row in rows_data:
    fl = codecs.open('part5Output.txt', 'ab', 'utf8')
    data_line = ','.join(row)
    fl.write(data_line + u'\r\n')
fl.close()
```

The output after running the code is below:





```
Rank,Player,Team,Points,Games,Avg
1,Cam Newton,CAR,389.1,16,24.3
2,Tom Brady,NE,343.7,16,21.5
3,Russell Wilson,SEA,336.4,16,21.0
4,Blake Bortles,JAC,316.1,16,19.8
5,Carson Palmer,FA,309.2,16,19.3
6,Drew Brees,NO,306.5,15,20.4
7,Aaron Rodgers,GB,301.3,16,18.8
8,Kirk Cousins,MIN,293.5,16,18.3
9,Matthew Stafford,DET,289.7,16,18.1
10,Eli Manning,NYG,287.6,16,18.0
11,Ryan Fitzpatrick,TB,285.1,16,17.8
12,Philip Rivers,LAC,284.3,16,17.8
13,Jameis Winston,TB,275.2,16,17.2
14,Derek Carr,OAK,273.3,16,17.1
15,Alex Smith,WAS,271.0,16,16.9
16,Tyrod Taylor,CLE,270.6,14,19.3
17,Ryan Tannehill,MIA,257.3,16,16.1
18,Andy Dalton,CIN,244.1,13,18.8
19,Matt Ryan,ATL,233.9,16,14.6
20,Ben Roethlisberger,PIT,227.6,12,19.0
21,Jay Cutler,FA,226.3,15,15.1
22,Marcus Mariota,TEN,210.1,12,17.5
23,Teddy Bridgewater,NO,200.4,16,12.5
24,Sam Bradford,ARI,194.8,14,13.9
25,Brian Hoyer,NE,166.6,11,15.1
26,Joe Flacco,BAL,162.1,10,16.2
27,Josh McCown,NYJ,132.1,8,16.5
28,Andrew Luck,IND,130.8,7,18.7
29,Blaine Gabbert,TEN,129.8,8,16.2
30,Brock Osweiler,MIA,116.7,8,14.6
31,Colin Kaepernick,FA,110.3,9,12.3
32,Nick Foles,PHI,95.9,11,8.7
33,Johnny Manziel,FA,95.0,9,10.6
34,Peyton Manning,FA,91.3,10,9.1
35,Matt Hasselbeck,FA,91.2,8,11.4
36,Ryan Mallett,FA,71.0,7,10.1
```

References:

- https://stackoverflow.com/questions/15137769/how-to-delete-same-words-from-different-text-file-using-python
- https://stackabuse.com/how-to-copy-a-file-in-python/
- https://docs.python.org/2/library/shutil.html
- https://www.sanfoundry.com/python-program-copy-contents-file-another/
- https://www.journaldev.com/14408/python-read-file-open-write-delete-copy
- https://unix.stackexchange.com/questions/145079/remove-all-lines-in-file-a-which-contain-the-strings-in-file-b