CSEE 5590

Python and Deep Learning

Lab 1 Report

- Harshini Medikonda
  Student ID: 18

CONTENTS:

INTRODUCTION:

This lab assignment familiarizes with the below concepts:

1. Sets
2. Lists
3. Loops and conditional statements
4. Object Oriented Concepts
5. Web page Parsing
6. Writing into files

OBJECTIVE:

These concepts are used in the below 5 tasks to provide solutions in the form of features.

- To search in a string and find the first non-repeated character in the given string
- Given two files, output the content of file 1 that is not in file2
- Given a list of students attending Python class and a list of students attending Web Application class, we have to find the list of students who are attending Python class and not Web Application class.
- To write a python program to create a hospital admission system
- To download webpage that contains a table using request library and then parse the page using Beautifulsoup library. Save all the information to the table

FEATURES:

The features in each task are:

Task 1: Find First Non-Repeated Character in a String

The user provides a string when prompted for an input. The given string consists of characters which may be repeated or non-repeated. Our goal is to find the first non-repeated character among all the characters.

Task 2: Contents of file 1 that are not in file 2

In the given files file 1 and file 2 and prints:

The content/words in file 1 that are not present in file 2. This is being written in to a new file.

Task 3: Find students attending Python but not Web Application

This task gives us two inputs. One is the list of students who are attending the Python class.

The second input gives the list of students attending Web Application class.

Among the students, our aim is to find the students that are attending Python class but not Web Application class.

Task 4: Design a hospital admission system

The task is to design a hospital admission system that has the below features.

1. Contains a book that allows to enter patient details
2. Patient name, age and the disease symptoms can be entered.
3. It is an admission system that stores the patient details sends it over to the nurse.
4. The default nurse is assigned to each patient
5. The doctor has the ability to change the nurse based on the patient’s disease.
6. The nurse, clerk and doctor details are printed once the admission is taken.

Task 5: Download a webpage and parse the table in it

The goal of this task is to download a webpage that contains a table and then parse it using Beautifulsoup library. Save all the table data into a file

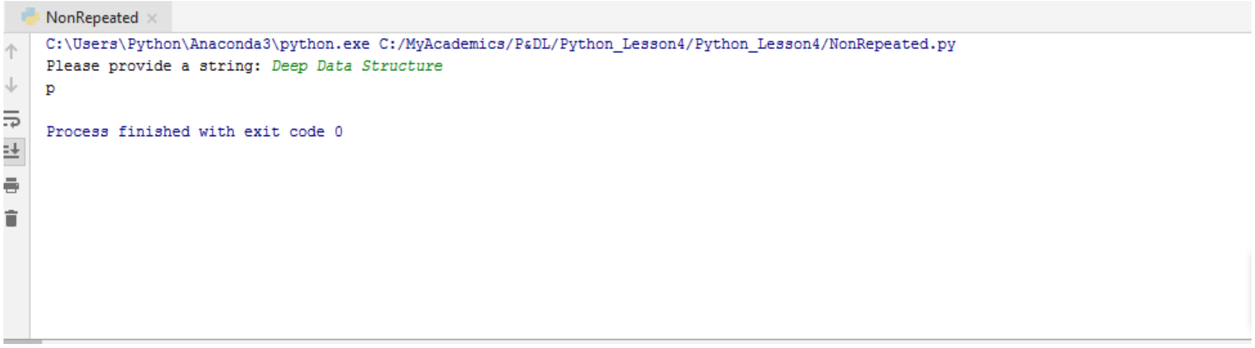
TOOLS:

PyCharm tool was used which uses Python 3 to execute the programs and get the output.

INPUT AND OUTPUT:

Task 1:

To find the first non-repeated character, the user provides a string as a input. This input is validated and the find the first non-repeated character is printed as output.

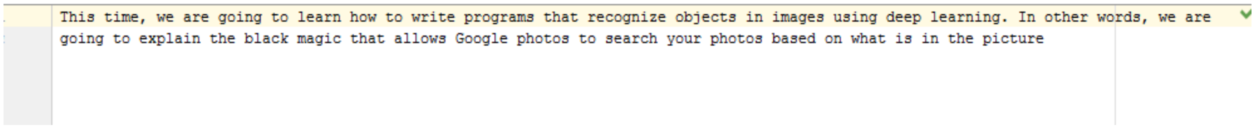


```
C:\Users\Python\Anaconda3\python.exe C:/MyAcademics/P&DL/Python_Lesson4/Python_Lesson4/NonRepeated.py
Please provide a string: Deep Data Structure
p

Process finished with exit code 0
```

Task 2:

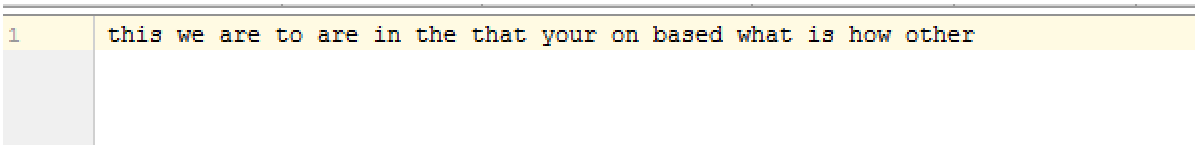
The task is remove the repeated content. The content present in file 1 is compared with the content in file 2 and then the non repeated content is written to new_file.txt.

Input file 1:



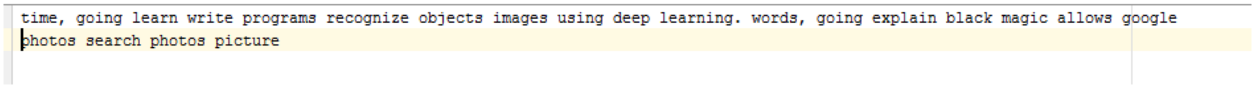
```
This time, we are going to learn how to write programs that recognize objects in images using deep learning. In other words, we are
going to explain the black magic that allows Google photos to search your photos based on what is in the picture
```

Input file 2:



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

Content in new_file.txt:



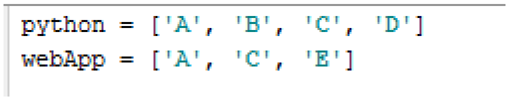
```
time, going learn write programs recognize objects images using deep learning. words, going explain black magic allows google
photos search photos picture
```

Task 3:

In this task, two inputs are given. First Input is a list of students attending Python class.

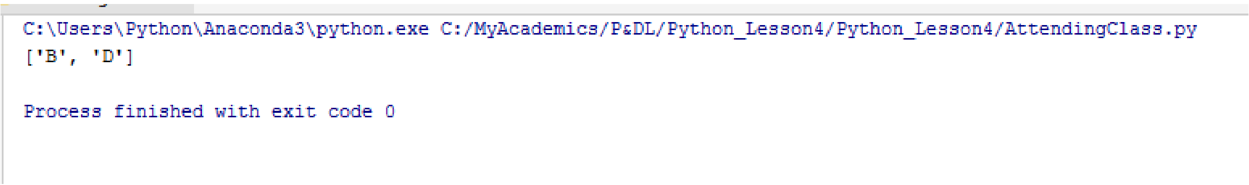
Second input is a list of students attending Web Application class.

Input :



```
python = ['A', 'B', 'C', 'D']
webApp = ['A', 'C', 'E']
```

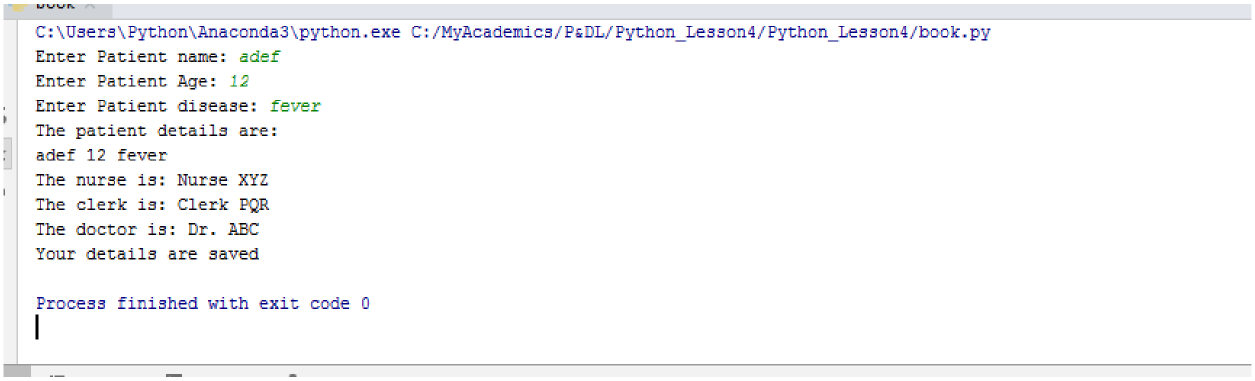
The below output is the list of students attending Python class but not Web Application



```
C:\Users\Python\Anaconda3\python.exe C:/MyAcademics/P&DL/Python_Lesson4/Python_Lesson4/AttendingClass.py
['B', 'D']

Process finished with exit code 0
```

Task 4:

The task is to build a hospital admission system that takes in patient details and assigns to a doctor, who can assign to a nurse and clerk to take care of the patient. The patient details are then saved to the system.



```
C:\Users\Python\Anaconda3\python.exe C:/MyAcademics/P&DL/Python_Lesson4/Python_Lesson4/book.py
Enter Patient name: adef
Enter Patient Age: 12
Enter Patient disease: fever
The patient details are:
adef 12 fever
The nurse is: Nurse XYZ
The clerk is: Clerk PQR
The doctor is: Dr. ABC
Your details are saved

Process finished with exit code 0
```

Task 5:

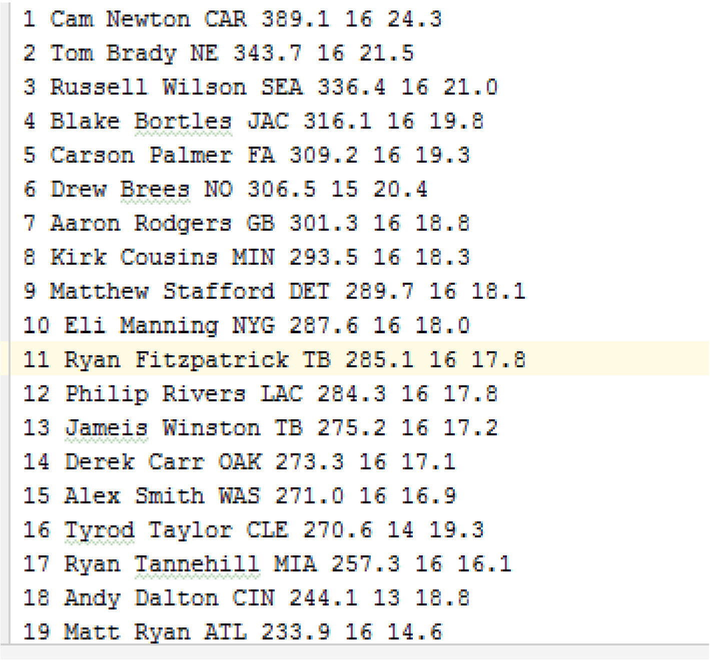
The task takes the web page as input and outputs the table contents into a file.

Input : Web page with the below URL

https://www.fantasypros.com/nfl/reports/leaders/qb.php?year=2015

Output:

The contents of the table are written to the file in the below format:

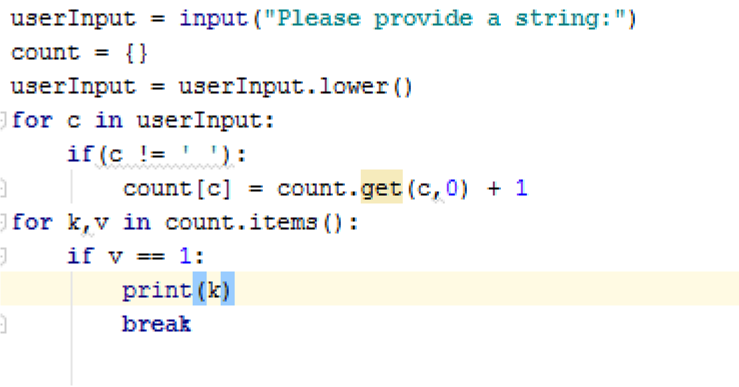


```
1 Cam Newton CAR 389.1 16 24.3
2 Tom Brady NE 343.7 16 21.5
3 Russell Wilson SEA 336.4 16 21.0
4 Blake Bortles JAC 316.1 16 19.8
5 Carson Palmer FA 309.2 16 19.3
6 Drew Brees NO 306.5 15 20.4
7 Aaron Rodgers GB 301.3 16 18.8
8 Kirk Cousins MIN 293.5 16 18.3
9 Matthew Stafford DET 289.7 16 18.1
10 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
```

IMPLEMENTATION:

Task 1:

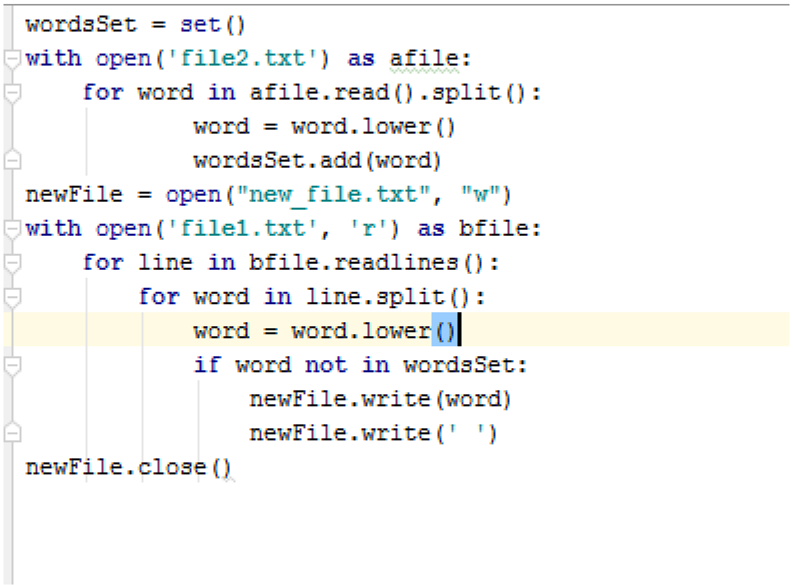
Here we use dictionary to store the count of each character. We increase the value once a value is repeated. At the end, the character with count 1 is displayed in the order of the characters given by the user.



```python
userInput = input("Please provide a string:")
count = {}
userInput = userInput.lower()
for c in userInput:
    if(c != ' '):
        count[c] = count.get(c,0) + 1
for k,v in count.items():
    if v == 1:
        print(k)
        break
```

Task 2:

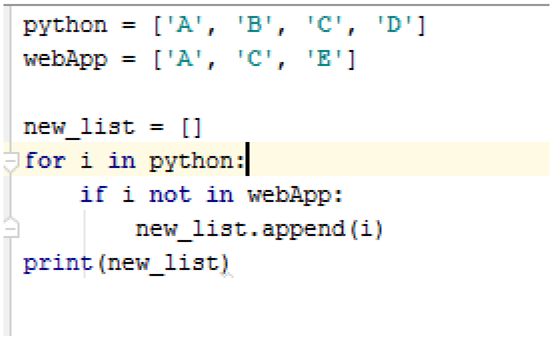
File 2 is read word by word and added to a set. Then file 1 is read word by word and if it is not present in the wordSet , then it is added to new_file.txt



```python
wordsSet = set()
with open('file2.txt') as afile:
    for word in afile.read().split():
            word = word.lower()
            wordsSet.add(word)
newFile = open("new_file.txt", "w")
with open('file1.txt', 'r') as bfile:
    for line in bfile.readlines():
        for word in line.split():
            word = word.lower()
            if word not in wordsSet:
                newFile.write(word)
                newFile.write(' ')
newFile.close()
```

Task 3:

Traverse though the list of students in python class and then look if that student is not present in web app class. If not present, then add to the list. Finally, output the list



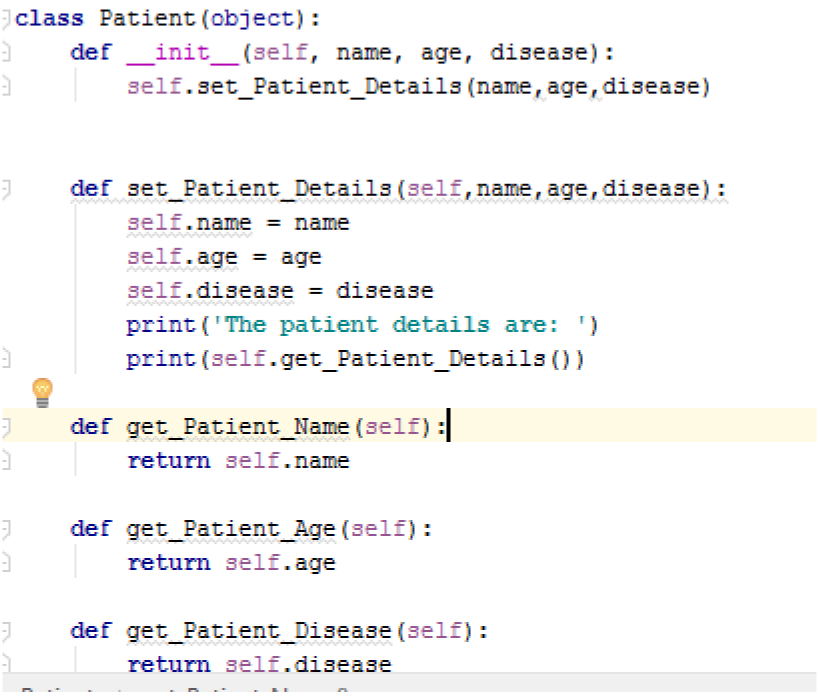
```
python = ['A', 'B', 'C', 'D']
webApp = ['A', 'C', 'E']

new_list = []
for i in python:
    if i not in webApp:
        new_list.append(i)
print(new_list)
```

Task 4:

There are 5 classes in the hospital admission system which implements below OOPS concepts

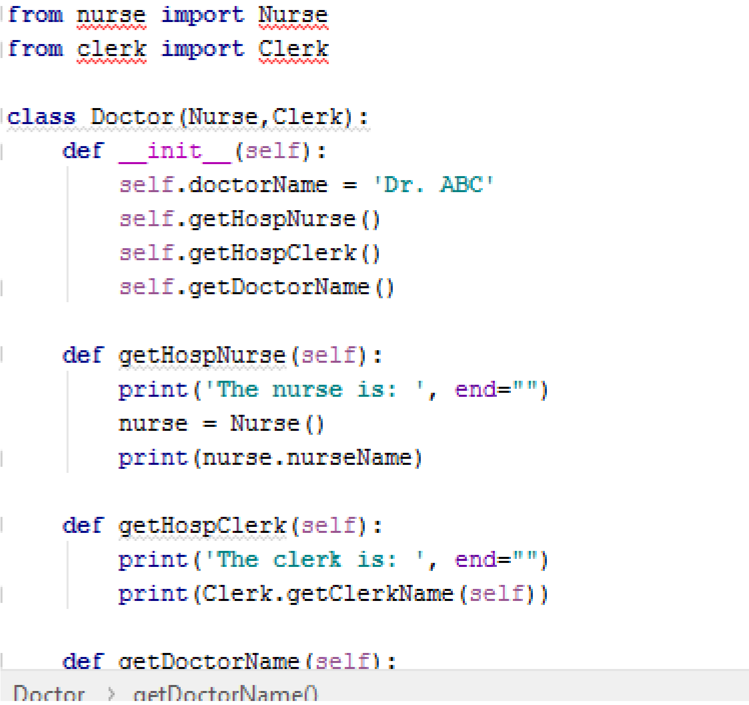
1. Each class has a constructor
2. Uses private variables
3. Uses static variables
4. Uses super and self
5. Uses multiple inheritance
6. Has super and sub classes
7. Implements most of the object oriented concepts

Patient Class:



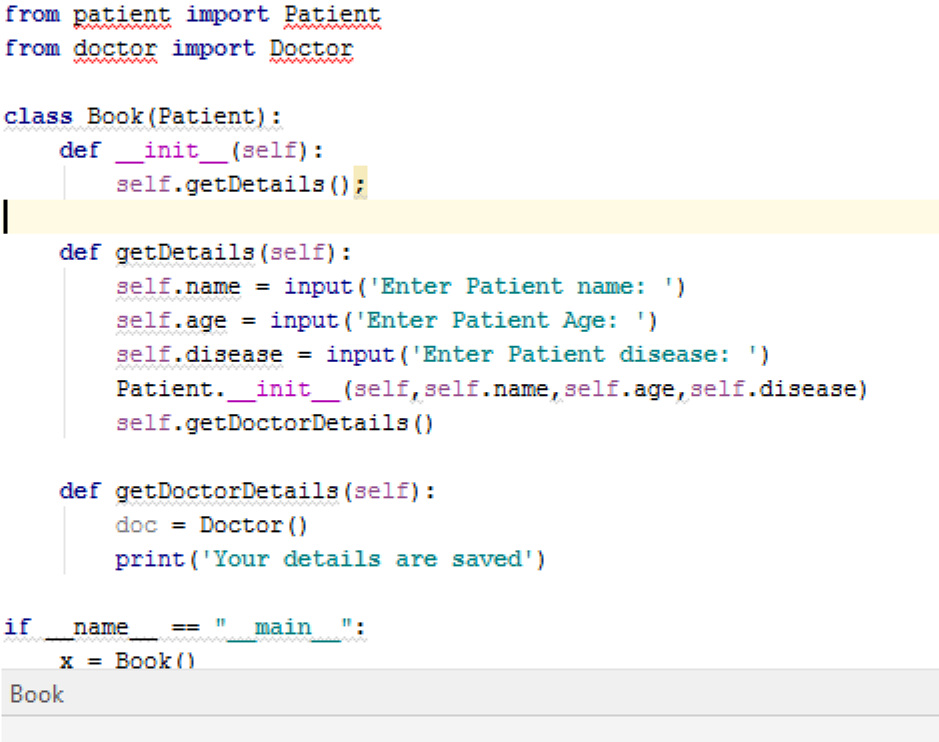
```
class Patient(object):
    def __init__(self, name, age, disease):
        self.set_Patient_Details(name,age,disease)


    def set_Patient_Details(self,name,age,disease):
        self.name = name
        self.age = age
        self.disease = disease
        print('The patient details are: ')
        print(self.get_Patient_Details())

    def get_Patient_Name(self):
        return self.name

    def get_Patient_Age(self):
        return self.age

    def get_Patient_Disease(self):
        return self.disease
```

Doctor.py



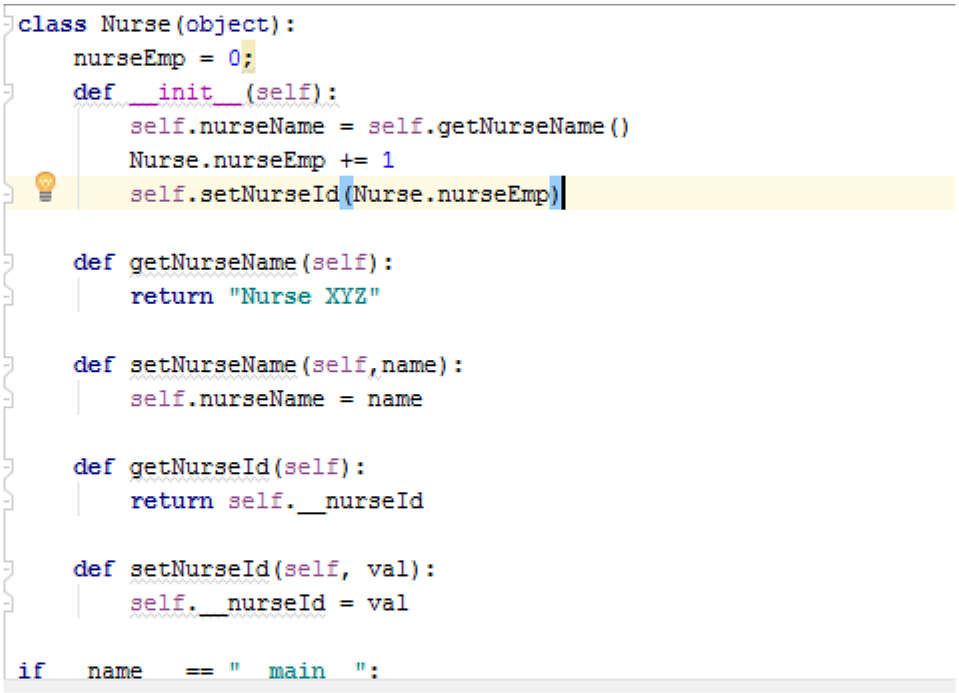
```
from nurse import Nurse
from clerk import Clerk

class Doctor(Nurse,Clerk):
    def __init__(self):
        self.doctorName = 'Dr. ABC'
        self.getHospNurse()
        self.getHospClerk()
        self.getDoctorName()

    def getHospNurse(self):
        print('The nurse is: ', end="")
        nurse = Nurse()
        print(nurse.nurseName)

    def getHospClerk(self):
        print('The clerk is: ', end="")
        print(Clerk.getClerkName(self))

    def getDoctorName(self):
```

Book.py



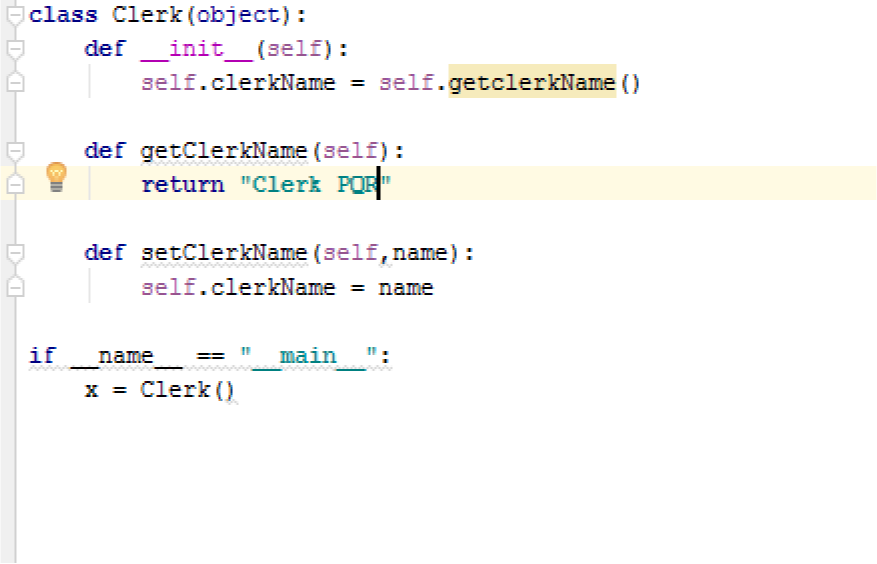
```
from patient import Patient
from doctor import Doctor

class Book(Patient):
    def __init__(self):
        self.getDetails();

    def getDetails(self):
        self.name = input('Enter Patient name: ')
        self.age = input('Enter Patient Age: ')
        self.disease = input('Enter Patient disease: ')
        Patient.__init__(self,self.name,self.age,self.disease)
        self.getDoctorDetails()

    def getDoctorDetails(self):
        doc = Doctor()
        print('Your details are saved')

if __name__ == "__main__":
    x = Book()
```

Nurse.py



```
class Nurse(object):
    nurseEmp = 0;
    def __init__(self):
        self.nurseName = self.getNurseName()
        Nurse.nurseEmp += 1
        self.setNurseId(Nurse.nurseEmp)

    def getNurseName(self):
        return "Nurse XYZ"

    def setNurseName(self,name):
        self.nurseName = name

    def getNurseId(self):
        return self.__nurseId

    def setNurseId(self, val):
        self.__nurseId = val

if __name__ == "__main__":
```

Clerk.py



```
class Clerk(object):
    def __init__(self):
        self.clerkName = self.getclerkName()

    def getClerkName(self):
        return "Clerk PQR"

    def setClerkName(self,name):
        self.clerkName = name

if __name__ == "__main__":
    x = Clerk()
```

RUNNING THE PROGRAM:

The code is being run in PyCharm that runs Python 3. The input and output screenshots are provided above.

REFERENCES:

https://www.geeksforgeeks.org/given-a-string-find-its-first-non-repeating-character/

https://www.geeksforgeeks.org/python-intersection-two-lists/

https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python

https://stackoverflow.com/questions/2010481/how-do-you-get-all-the-rows-from-a-particular-table-using-beautifulsoup