Lab Assignment-1

Lab-1 Python and Deep learning

Name of the instructor: Dr. Lee

Lab ID-6

Team Members:

1.Tarun Teja Kasturi (Class ID-11)

2.Lakshmana Kumar Mettu (Class ID-16)

3.Pavan Kumar Chongala (Class ID-04)

Objective: The main objective of this lab-1 is to know about basic concepts of python such as tuples,Object oriented concepts,web scraping,Union etc.

Task-1:To compute the net amount of a bank account based a transaction log from console input.

For this first make net amount to be zero and using conditional statements if user make deposit then it would be added to net amount otherwise if it is withdraw then amount would be reduced.The Corresponding Code snippet along with output shown below.

Finding of net bank amount:

Lab Assignment-1

```python
# define a variable for main amount
remaining_amount = 0

while True:
    # input the transaction
    str = input ("Enter transaction: ")

    # get the value type and amount to the list
    # seprated by space
    transaction = str.split(" ")

    # get the value of transaction type and amount
    # in the separated variables
    type = transaction[0]
    amount = int(transaction[1])

    if type=="Deposit" or type=="deposit":
        remaining_amount += amount
    elif type=="Withdraw" or type=="withdraw":
        remaining_amount -= amount
    else:
        pass

    #input choice
    str = input ("want to continue (Y for yes) : ")
    if not (str[0] =="Y" or str[0] =="y") :
        # break the loop
        break

# print the remaining amount
print ("Remaining amount: ", remaining_amount)
```

Output of program-1

```
C:\Users\laksh\PycharmProjects\pythonlab\venv\Scripts\python.exe
Enter transaction: Deposit 1000
want to continue (Y for yes) : y
Enter transaction: Withdraw 200
want to continue (Y for yes) : y
Enter transaction: Deposit 100
want to continue (Y for yes) : n
Remaining amount:   900
```

Task-2:To create a dictionary with keys as names and values as list of (subjects, marks) in sorted order.

Suppose If we have a list of tuples as follows:[('John', ('Physics', 80)) , (' Daniel', ('Science', 90)), ('John', ('Science', 95)), ('Mark',('Maths', 100)), ('Daniel', ('History', 75)), ('Mark', ('Social', 95))] using tups and dict functions we will get the required order.The screenshots with proper comments are given below.

Code

```python
def Convert(tup, di):  #Convert a list of Tuples into Dictionary
    for a, b in tup:
        di.setdefault(a, []).append(b)
    return di


# Driver Code
tups = [( 'John', ('Physics', 80)),('Daniel', ('Science',90)),('John', ('Science', 95)),
        ('Mark',('Maths', 100)), ('Daniel', ('History', 75)), ('Mark', ('Social', 95))]
dictionary = {}#for sorting of given tuple
print(Convert(tups, dictionary))
```

Output

```
C:\Users\laksh\PycharmProjects\pythonlab\venv\Scripts\python.exe C:/Users/laksh/PycharmProjects/pythonlab/que2.py
{'John': [('Physics', 80), ('Science', 95)], 'Daniel': [('Science', 90), ('History', 75)], 'Mark': [('Maths', 100), ('Social', 95)]}

Process finished with exit code 0
```

Task-3:Consider the following scenario. You have a list of students who are attending class "Python" and

Lab Assignment-1

another list of students who are attending class "Web Application".To find the list of students who are attending both the classes.Also find the list of students who are not common in both the classes.

This can be done using union function.The appropriate output with outcome has given below.

Program-3

```python
P = input("enter names of students in python class:")
Python = set(P.split(" ")) # input get sepearted by spaces between them
#print(Python)

# list of students who took web
W = input("enter names of students in Web class:")
web = set(W.split(" "))
#print(Web)


# printing the common students who took both python and web using & operator
print("printing the common students who took both python and web")
print(Python & web)


# storing the list of students who took only python to the variable 'a'
a = Python-web

# storing the list of students who took only web to the variable 'a'
b = web-Python

# printing the list of students who are not in common in both python and web
print("printing the list of students who are not in common in both python and web:")
print(a.union(b))
```

Outcome

```
C:\Users\laksh\PycharmProjects\pythonlab\venv\Scripts\python.exe C:/Users/laksh/PycharmProjects/pythonlab/que3.py
enter names of students in python class:lakshman tarun pavan
enter names of students in Web class:anil tarun praneeth
printing the common students who took both python and web
{'tarun'}
printing the list of students who are not in common in both python and web:
{'praneeth', 'lakshman', 'pavan', 'anil'}
```

Task-4: To get the longest substring without repeating characters along with the length.

In order to implement this one We are using max and len functions.

Code with comments

```python
word=input('Enter the string to be checked:')

currentLength = 1 #intially current length is equal to one
longest = 1
last_seen = {word[0] : 0}
i = 1

while i < len(word):
    letter = word[i]
    if letter in last_seen:
        i = last_seen[letter] + 1
        last_seen.clear()
        longest = max(longest, currentLength) # if the last seen and the current letter are same then length becomes minus one
        currentLength = 0
    else:
        last_seen[letter] = i # if the last seen is not same with cureent letter then current length incremented by one.
        currentLength += 1
    i += 1

longest = max(longest, currentLength) #using the max function we wre getting the longest substring as longest and length as current length
print('The longest substring is:\n',word[:longest],'\nLength of sub String is:\n',longest)
```

Output

```
C:\Users\laksh\PycharmProjects\pythonlab\venv\Scripts\python.exe C:/Users/laksh/PycharmProjects/pythonlab/que4.py
Enter the string to be checked:abcabc
The longest substring is:
 abc
Length of sub String is:
 3
```

Task-5: To create Airline Booking Reservation System (e.g. classes Flight, Person, Employee, Passenger etc.)

The code contains five classes,*init* constructor in all the classes,inheritance,super call,usage of self,private data member,multiple inheritance along with instances of all corresponding classes.Comments along with code as given below.

Program

Lab Assignment-1

```python
class Person:  #base class
    per_count=0
    def __init__(self,name,email): #default constructor
        self.name=name
        self.email=email
        Person.per_count+=1
    def DisplayDetails(self):
        print("Name: ",self.name,"Email: ",self.email)
    def DispCount(self):
        print("Total count of persons: ",Person.per_count)

class Employee(Person):#Inheritance Usage
    def __init__(self,Emp_name,email,Employee_phone): #default init constructor
        Person.__init__(self,Emp_name,email)
        self.Employee_phone = Employee_phone
        self.Employee_phone=Employee_phone
    def DisplayDetails(self):
        Person.DisplayDetails(self)
        print("Employee_phone: ",self.Employee_phone)

class Passenger(Employee): #inheritance
    def __init__(self,name,email,emp_phone,Emp_name,e_email,Employee_phone,time,date): #default Init constructor
        super().__init__(name,email,Employee_phone)   #usage of super call
        self.emp_phone=emp_phone
        self.name=name
        self.email=email
        self.ename=Emp_name
        self.eemail=e_email
        self.ephone=Employee_phone
        self.time=time
        self.date=date
    def DisplayDetails(self):
        Person.DisplayDetails(self)
        print("Emp phone:", self.emp_phone)
        print("The Airplane time is: ",self.time)
        print("The Reserved date is: ",self.date)
```

Lab Assignment-1

```python
class Book():
    def __init__(self,per_name,Travel_date, travel_time): #default init constructor
        self.person_name=per_name
        self.travel=Travel_date
        self.time=travel_time
    def DisplayDetaails(self):
        print("Person_name: ",self.person_name,"Travel date: ",self.travel,"Travel time: ",self.ti:

class Flight(Person,Book): #multiple Inheritance
    def __init__(self,name,email,per_name,Travel_date,travel_time): #default init constructor
        # super().__init__(name,email)
        Person.__init__(self,name,email)
        Book.__init__(self,per_name,Travel_date, travel_time)

    def DisplayDetails(self):
        Book.DisplayDetaails(self)
        Person.DisplayDetails(self)


    #creation of instances for above classes

person1=Person('Lakshman','lakshman95@gmail.com')
person2=Person('Praneeth','praneeththota@gmail.com')
Employee1=Employee('Tarun','tarunkasturi@gmail.com','9162628454')
Employee2=Employee('Pavan','pavan123@gmail.com','916454585')
Book1=Book('Lakshman','15 march 2019','12 am')
Book2=Book('Praneeth','20 April 2019','2 pm')
print("##Persons Count##")
Person.DispCount(Person)

#Flight attending persons
Flight1=Flight('Lakshman','lakshman95@gmail.com','lakshman','15 march 2019', '12am')
Flight2=Flight('Praneeth','praneeththota@gmail.com','praneeth','20 April 2019','2 pm')

print("###Employee Details###")
Employee1.DisplayDetails();
Employee2.DisplayDetails();
```

Lab Assignment-1

```python
#Flight attending persons
Flight1=Flight('Lakshman','lakshman95@gmail.com','lakshman','15 march 2019', '12am')
Flight2=Flight('Praneeth','praneeththota@gmail.com','praneeth','20 April 2019','2 pm')

print("###Employee Details###")
Employee1.DisplayDetails();
Employee2.DisplayDetails();

print("##Person Details##")
person1.DisplayDetails();
person2.DisplayDetails();

print("##Book Details##")
Book1.DisplayDetaails();
Book2.DisplayDetaails();

print("#Flight attending person details")
Flight1.DisplayDetaails();


Passenger1=Passenger("brundha","brundha@gmail.com","9162625252","Tarun","tarunkasturi@gmail.com","12345","9 40 PM","16 JUNE")
print("passenger Appointments-1")
Passenger1.DisplayDetails()

Passenger2=Passenger("Yasritha","yasritha@gmail.com","990909876","pavan","pavan123@gmail.com","898976","8 00 AM","21 JUNE")
print("Passenger Appointments-2")
Passenger2.DisplayDetails()
```

Outputs

```
C:\Users\laksh\PycharmProjects\pythonlab\venv\Scripts\python.exe C:/Users/laksh/PycharmProjects/pythonlab/que5.py
##Persons Count##
Total count of persons:  4
###Employee Details###
Name:  Tarun Email:  tarunkasturi@gmail.com
Employee_phone:  9162628454
Name:  Pavan Email:  pavan123@gmail.com
Employee_phone:  916454585
##Person Details##
Name:  Lakshman Email:  lakshman95@gmail.com
Name:  Praneeth Email:  praneeththota@gmail.com
```

```
##Book Details##
Person_name:  Lakshman Travel date:  15 march 2019 Travel time:  12 am
Person_name:  Praneeth Travel date:  20 April 2019 Travel time:  2 pm
#Flight attending person details
Person_name:  lakshman Travel date:  15 march 2019 Travel time:  12am
passenger Appointments-1
Name:  brundha Email:  brundha@gmail.com
Emp phone: 9162625252
The Airplane time is:  9 40 PM
The Reserved date is:  16 JUNE
Passenger Appointments-2
```

```
 The Airplane time is:  9 40 PM
 The Reserved date is:  16 JUNE
 Passenger Appointments-2
 Name:  Yasritha Email:  yasritha@gmail.com
 Emp phone: 990909876
 The Airplane time is:  8 00 AM
 The Reserved date is:  21 JUNE
```

Task-6:Program a code which download a webpage contains a table using Request library, then parse the page using Beautiful soup library.

It should save the information about the states and their capitals in a file. Here it is saved as .txt file.Code with comments as given below.

Program

Lab Assignment-1

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


url = "https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States"
fpURL= urllib.request.urlopen(url)

#parsing the givenhtml page
soup = BeautifulSoup(fpURL, "html.parser")

#finfing table im the page

table = soup.find("table")

#finding table headers from the page
table_header = table.find_all('th')

#iterating through the headers and displaying the headers in text format
th = [t.text for t in table_header]
print(th)

#saving the output to a file
file = open("output.txt", "w")

#writing the table headers to the file
file.write(str(th))

#finding the table rows in the tabel
tables_row = table.find_all("tr")

for tr in tables_row:    #for each row in the table
    td = tr.find_all('td')    #finfing all the table cells in the table, td stands for table cells
    row = [r.text for r in td]    #displaying each row and cell in text format
    file.write("\n")  #wrrting each row in new line
    print(row)
    file.write(str(row))  #writing the output
```

IDE and Plugin Updates
PyCharm is ready to update.

Outputs

```
['Name &postal abbreviation[12]\n', 'Cities\n', 'Established[C]\n', 'Population[D][14]\n', 'Total area[15]\n', 'Land area[15]\n', 'Water area[15]\n
[]
[]
['AL\n', 'Montgomery\n', 'Birmingham\n', 'Dec 14, 1819\n', '4,874,747\n', '52,420\n', '135,767\n', '50,645\n', '131,171\n', '1,775\n', '4,597\n', '7
['AK\n', 'Juneau\n', 'Anchorage\n', 'Jan 3, 1959\n', '739,795\n', '665,384\n', '1,723,337\n', '570,641\n', '1,477,953\n', '94,743\n', '245,384\n', '
['AZ\n', 'Phoenix\n', 'Feb 14, 1912\n', '7,016,270\n', '113,990\n', '295,234\n', '113,594\n', '294,207\n', '396\n', '1,026\n', '9\n']
['AR\n', 'Little Rock\n', 'Jun 15, 1836\n', '3,004,279\n', '53,179\n', '137,732\n', '52,035\n', '134,771\n', '1,143\n', '2,961\n', '4\n']
['CA\n', 'Sacramento\n', 'Los Angeles\n', 'Sep 9, 1850\n', '39,536,653\n', '163,695\n', '423,967\n', '155,779\n', '403,466\n', '7,916\n', '20,501\n',
['CO\n', 'Denver\n', 'Aug 1, 1876\n', '5,607,154\n', '104,094\n', '269,601\n', '103,642\n', '268,431\n', '452\n', '1,170\n', '7\n']
['CT\n', 'Hartford\n', 'Bridgeport\n', 'Jan 9, 1788\n', '3,588,184\n', '5,543\n', '14,357\n', '4,842\n', '12,542\n', '701\n', '1,816\n', '5\n']
['DE\n', 'Dover\n', 'Wilmington\n', 'Dec 7, 1787\n', '961,939\n', '2,489\n', '6,446\n', '1,949\n', '5,047\n', '540\n', '1,399\n', '1\n']
['FL\n', 'Tallahassee\n', 'Jacksonville\n', 'Mar 3, 1845\n', '20,984,400\n', '65,758\n', '170,312\n', '53,625\n', '138,887\n', '12,133\n', '31,424\n
['GA\n', 'Atlanta\n', 'Jan 2, 1788\n', '10,429,379\n', '59,425\n', '153,910\n', '57,513\n', '148,959\n', '1,912\n', '4,951\n', '14\n']
['HI\n', 'Honolulu\n', 'Aug 21, 1959\n', '1,427,538\n', '10,932\n', '28,313\n', '6,423\n', '16,635\n', '4,509\n', '11,678\n', '2\n']
['ID\n', 'Boise\n', 'Jul 3, 1890\n', '1,716,943\n', '83,569\n', '216,443\n', '82,643\n', '214,045\n', '926\n', '2,398\n', '2\n']
['IL\n', 'Springfield\n', 'Chicago\n', 'Dec 3, 1818\n', '12,802,023\n', '57,914\n', '149,995\n', '55,519\n', '143,793\n', '2,395\n', '6,202\n', '18\n
['IN\n', 'Indianapolis\n', 'Dec 11, 1816\n', '6,666,818\n', '36,420\n', '94,326\n', '35,826\n', '92,789\n', '593\n', '1,537\n', '9\n']
['IA\n', 'Des Moines\n', 'Dec 28, 1846\n', '3,145,711\n', '56,273\n', '145,746\n', '55,857\n', '144,669\n', '416\n', '1,077\n', '4\n']
['KS\n', 'Topeka\n', 'Wichita\n', 'Jan 29, 1861\n', '2,913,123\n', '82,278\n', '213,100\n', '81,759\n', '211,754\n', '520\n', '1,346\n', '4\n']
['KY\n', 'Frankfort\n', 'Louisville\n', 'Jun 1, 1792\n', '4,454,189\n', '40,408\n', '104,656\n', '39,486\n', '102,269\n', '921\n', '2,387\n', '6\n']
['LA\n', 'Baton Rouge\n', 'New Orleans\n', 'Apr 30, 1812\n', '4,684,333\n', '52,378\n', '135,659\n', '43,204\n', '111,898\n', '9,174\n', '23,761\n',
['ME\n', 'Augusta\n', 'Portland\n', 'Mar 15, 1820\n', '1,335,907\n', '35,380\n', '91,633\n', '30,843\n', '79,883\n', '4,537\n', '11,750\n', '2\n']
['MD\n', 'Annapolis\n', 'Baltimore\n', 'Apr 28, 1788\n', '6,052,177\n', '12,406\n', '32,131\n', '9,707\n', '25,142\n', '2,699\n', '6,990\n', '8\n']
['MA\n', 'Boston\n', 'Feb 6, 1788\n', '6,859,819\n', '10,554\n', '27,336\n', '7,800\n', '20,202\n', '2,754\n', '7,134\n', '9\n']
['MI\n', 'Lansing\n', 'Detroit\n', 'Jan 26, 1837\n', '9,962,311\n', '96,714\n', '250,487\n', '56,539\n', '146,435\n', '40,175\n', '104,052\n', '14\n
['MN\n', 'St. Paul\n', 'Minneapolis\n', 'May 11, 1858\n', '5,576,606\n', '86,936\n', '225,163\n', '79,627\n', '206,232\n', '7,309\n', '18,930\n', '8\
```

References:

1. https://stackoverflow.com/questions/986006/how-do-i-pass-a-variable-by-reference

Lab Assignment-1