Python Lab_Assignment-1

> ***Authors***

Lab Id-1

1.Nagababu Chilukuri, Class Id-26

2.Nikita Goyal, Class Id-7

3.Ronnie Dean Antwiler II, Class Id-1

Introduction:

A lab Assignment was completed by using the concepts and logics taught in the class like Classes,sets,tuples,dictionary and many more.

Objective:

The main Objective is to understand and learn the following concepts of Python Programming:

1. Sets, Tuples, dictionary
2. lists,classes,objects,inheritance
3. Beautiful Soup

Workflow:

Question-1:

Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:

Suppose the following input is supplied to the program:

Deposit 300

Deposit 250

Withdraw 100

Deposit 50

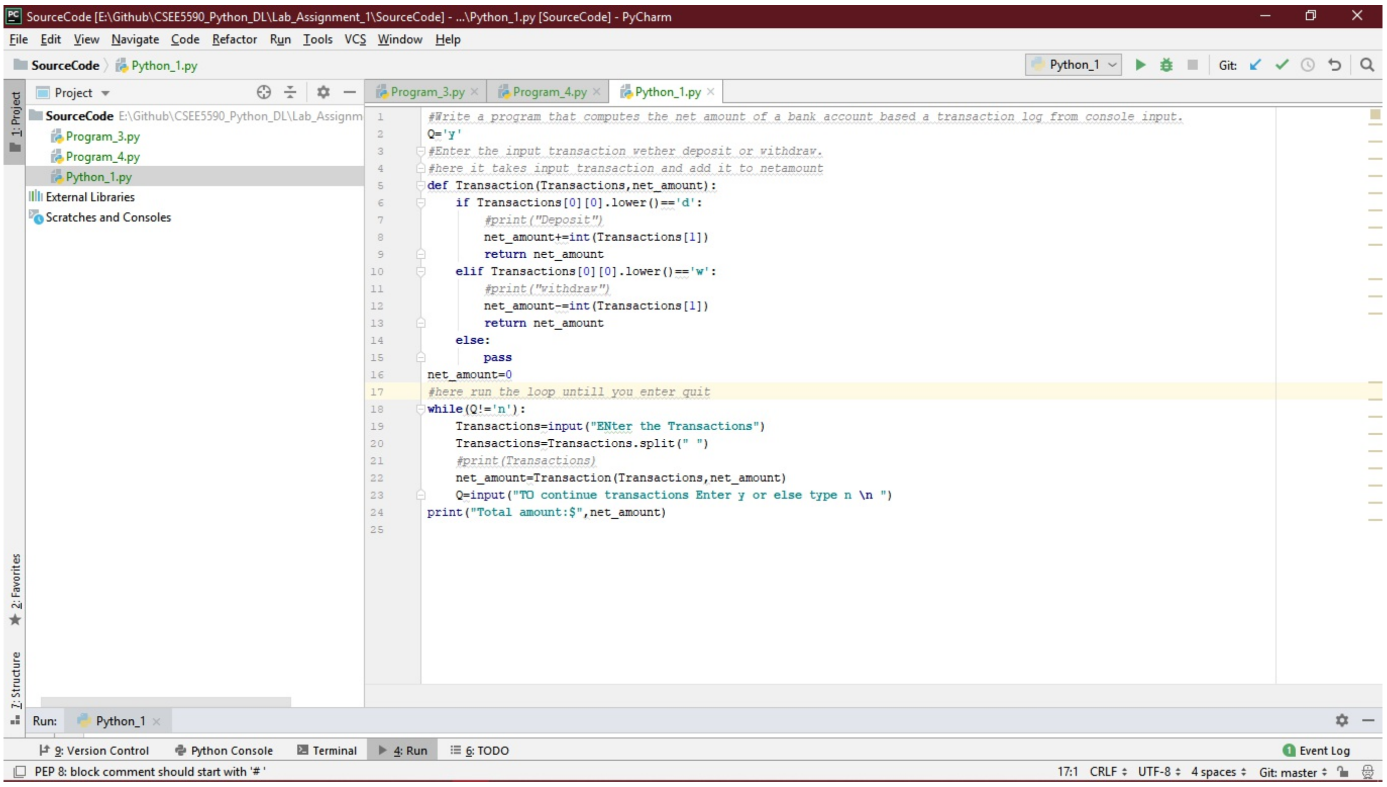
Then the output should be Total amount -$500

Solution-1

Implementation

The string input transation is taken from console then pass it to the function. The function calculates whether to increment the net amount when Deposit input is given, otherwise decrement the net amount when Wirthdraw input is given.

> *Code Snippet:-*



```python
#Write a program that computes the net amount of a bank account based a transaction log from console input.
Q='y'
#Enter the input transaction wether deposit or withdraw.
#here it takes input transaction and add it to netamount
def Transaction(Transactions,net_amount):
    if Transactions[0][0].lower()=='d':
        #print("Deposit")
        net_amount+=int(Transactions[1])
        return net_amount
    elif Transactions[0][0].lower()=='w':
        #print("withdraw")
        net_amount-=int(Transactions[1])
        return net_amount
    else:
        pass
net_amount=0
#here run the loop untill you enter quit
while(Q!='n'):
    Transactions=input("ENter the Transactions")
    Transactions=Transactions.split(" ")
    #print(Transactions)
    net_amount=Transaction(Transactions,net_amount)
    Q=input("TO continue transactions Enter y or else type n \n ")
print("Total amount:$",net_amount)
```

PEP 8: block comment should start with '# '

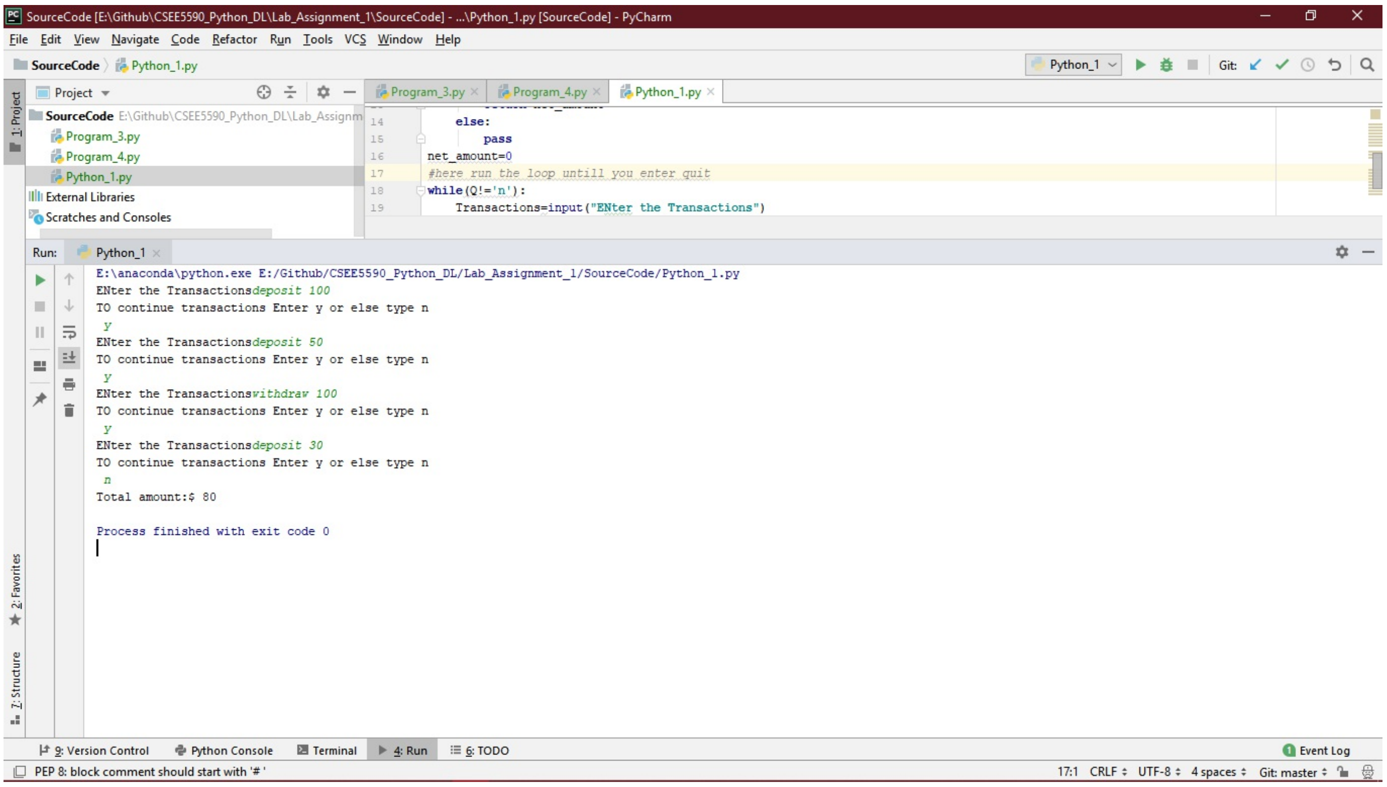
> *Input and Output:-*

Enter input Transaction either Deposit or Withdraw.

> ***Exapmle Input:-***

Deposit 100

Withdraw 50



```
    else:
        pass
net_amount=0
#here run the loop untill you enter quit
while(Q!='n'):
    Transactions=input("ENter the Transactions")
```

```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Python_1.py
ENter the Transactionsdeposit 100
TO continue transactions Enter y or else type n
 y
ENter the Transactionsdeposit 50
TO continue transactions Enter y or else type n
 y
ENter the Transactionswithdraw 100
TO continue transactions Enter y or else type n
 y
ENter the Transactionsdeposit 30
TO continue transactions Enter y or else type n
 n
Total amount:$ 80

Process finished with exit code 0
```

SourceCode [E:\Github\CSEE5590_Python_DL\Lab_Assignment_1\SourceCode] - ...\Python_1.py [SourceCode] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
SourceCode Python_1.py
Program_3.py Program_4.py Python_1.py
External Libraries
Scratches and Consoles
PEP 8: block comment should start with '# '

Question-2:

Suppose you 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))]

Create a dictionary with keys as names and values as list of (subjects, marks) in sorted order.

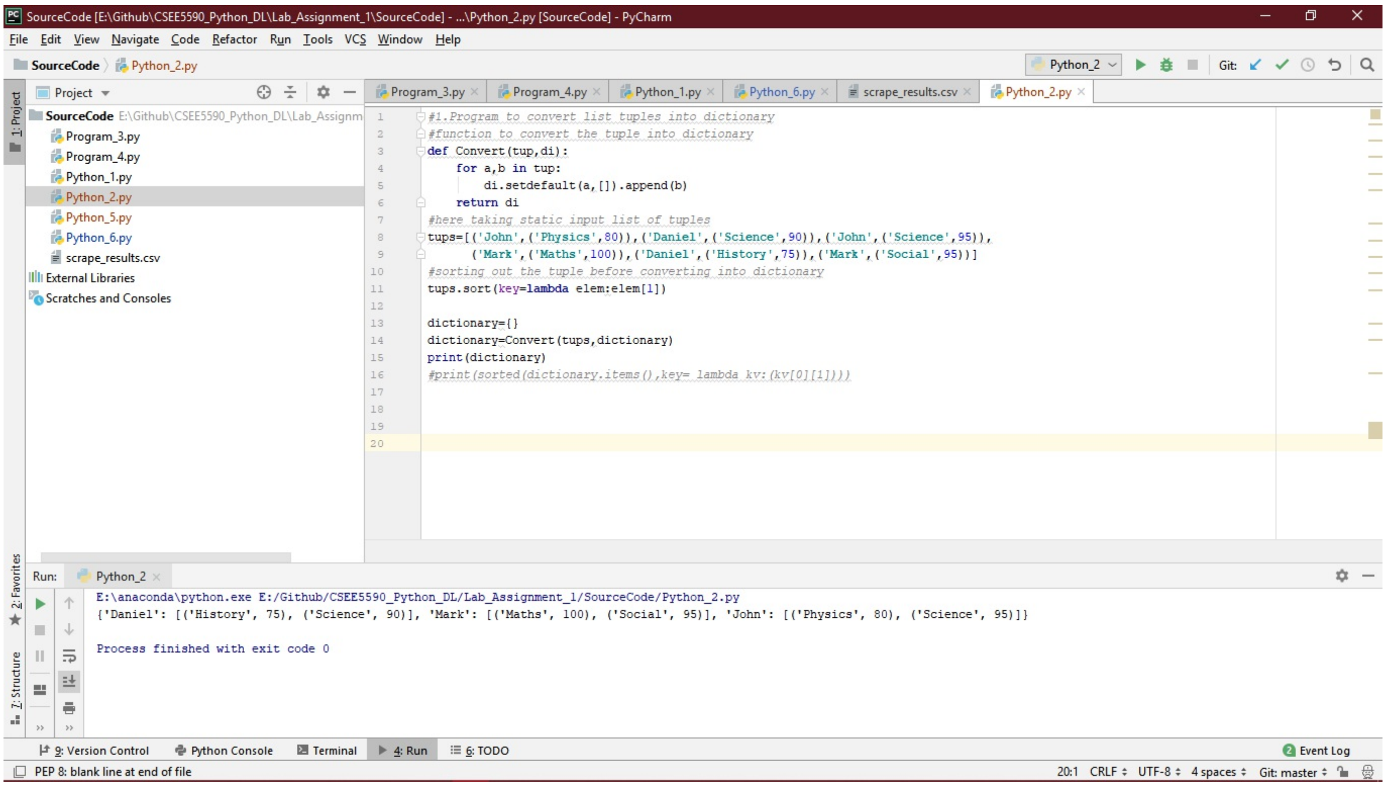
{John : [(‘Physics’, 80), (‘Science’, 95)]Daniel : [(’History’, 75), (‘Science’, 90)]Mark : [(‘Maths’, 100), (‘Social’, 95)]}

Solution-2:

> ***Implementation:-***

Inorder to get list of tuples into dictionary, first initialize the tuple list and convert into sorted order with subjects and marks tuple and then convert it into dictionary.

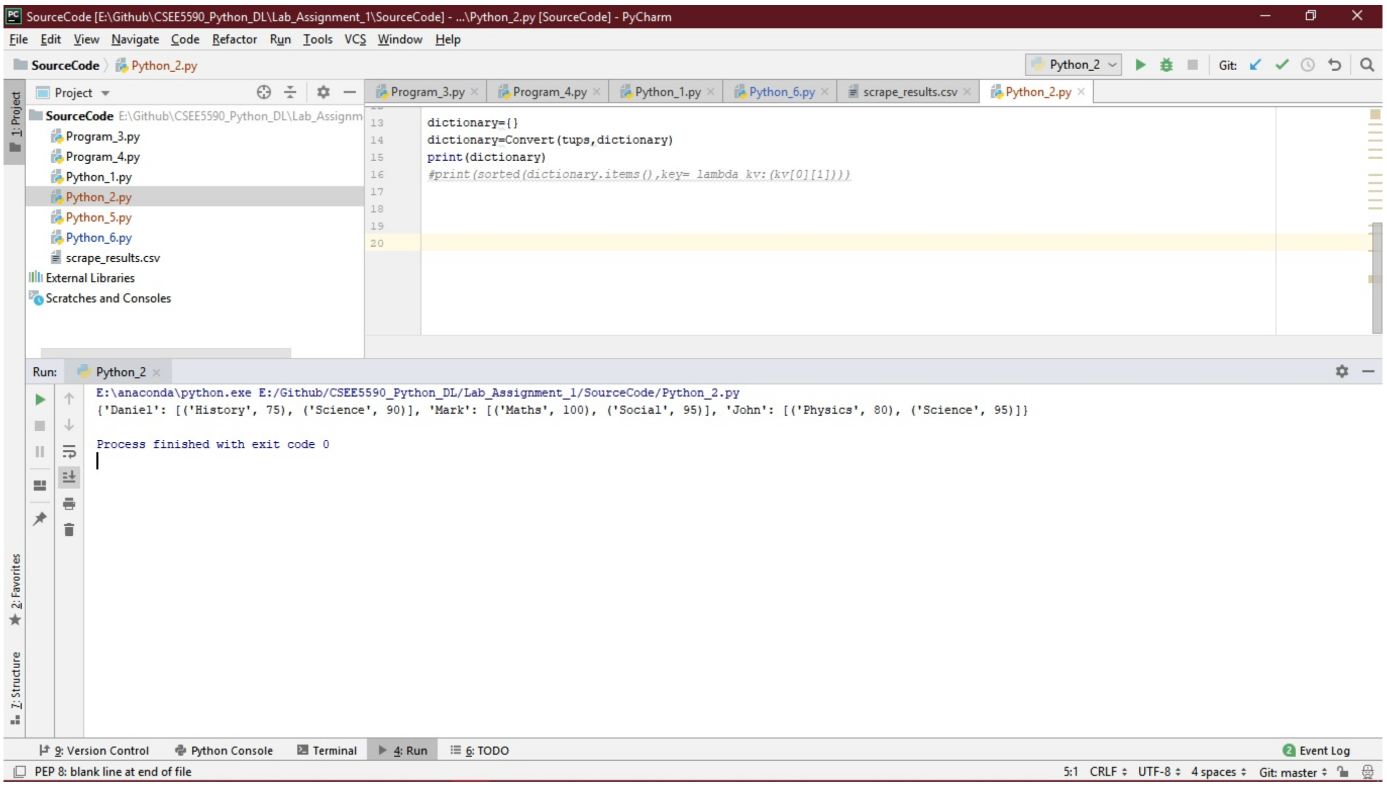
Code Snippet:



```python
#1.Program to convert list tuples into dictionary
#function to convert the tuple into dictionary
def Convert(tup,di):
    for a,b in tup:
        di.setdefault(a,[]).append(b)
    return di
#here taking static input list of tuples
tups=[('John',('Physics',80)),('Daniel',('Science',90)),('John',('Science',95)),
      ('Mark',('Maths',100)),('Daniel',('History',75)),('Mark',('Social',95))]
#sorting out the tuple before converting into dictionary
tups.sort(key=lambda elem:elem[1])

dictionary={}
dictionary=Convert(tups,dictionary)
print(dictionary)
#print(sorted(dictionary.items(),key= lambda kv:(kv[0][1])))
```

```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Python_2.py
{'Daniel': [('History', 75), ('Science', 90)], 'Mark': [('Maths', 100), ('Social', 95)], 'John': [('Physics', 80), ('Science', 95)]}

Process finished with exit code 0
```

Input and Output:



```
dictionary={}
dictionary=Convert(tups,dictionary)
print(dictionary)
#print(sorted(dictionary.items(),key= lambda kv:(kv[0][1])))
```

```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Python_2.py
{'Daniel': [('History', 75), ('Science', 90)], 'Mark': [('Maths', 100), ('Social', 95)], 'John': [('Physics', 80), ('Science', 95)]}

Process finished with exit code 0
```

Question-3:

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”.

1)Find the list of students who are attending both the classes.

2)Also find the list of students who are not common in both the classes.

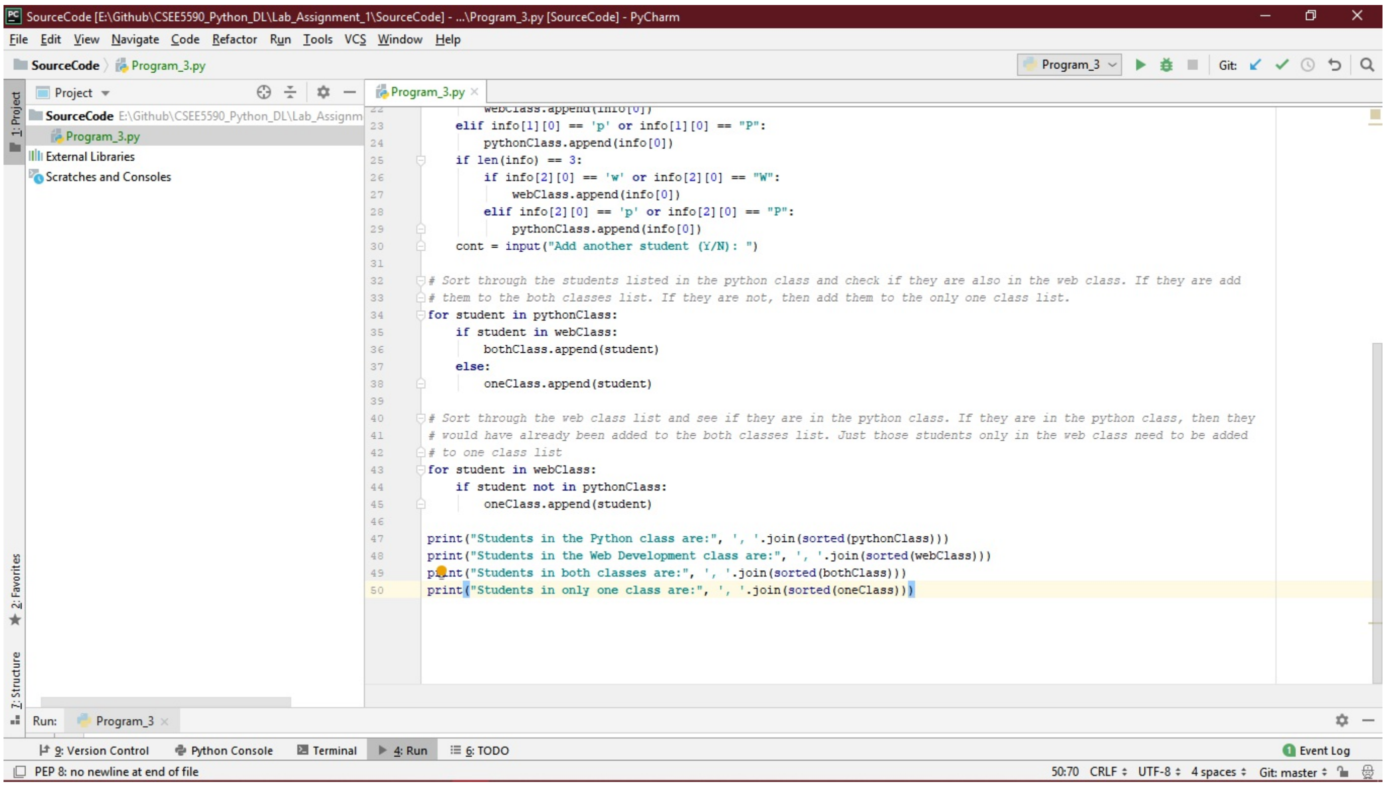
Print the both lists. Consider accepting the input from the console for list of students that belong to class “Python” and class “Web Application”.

Solution-3:

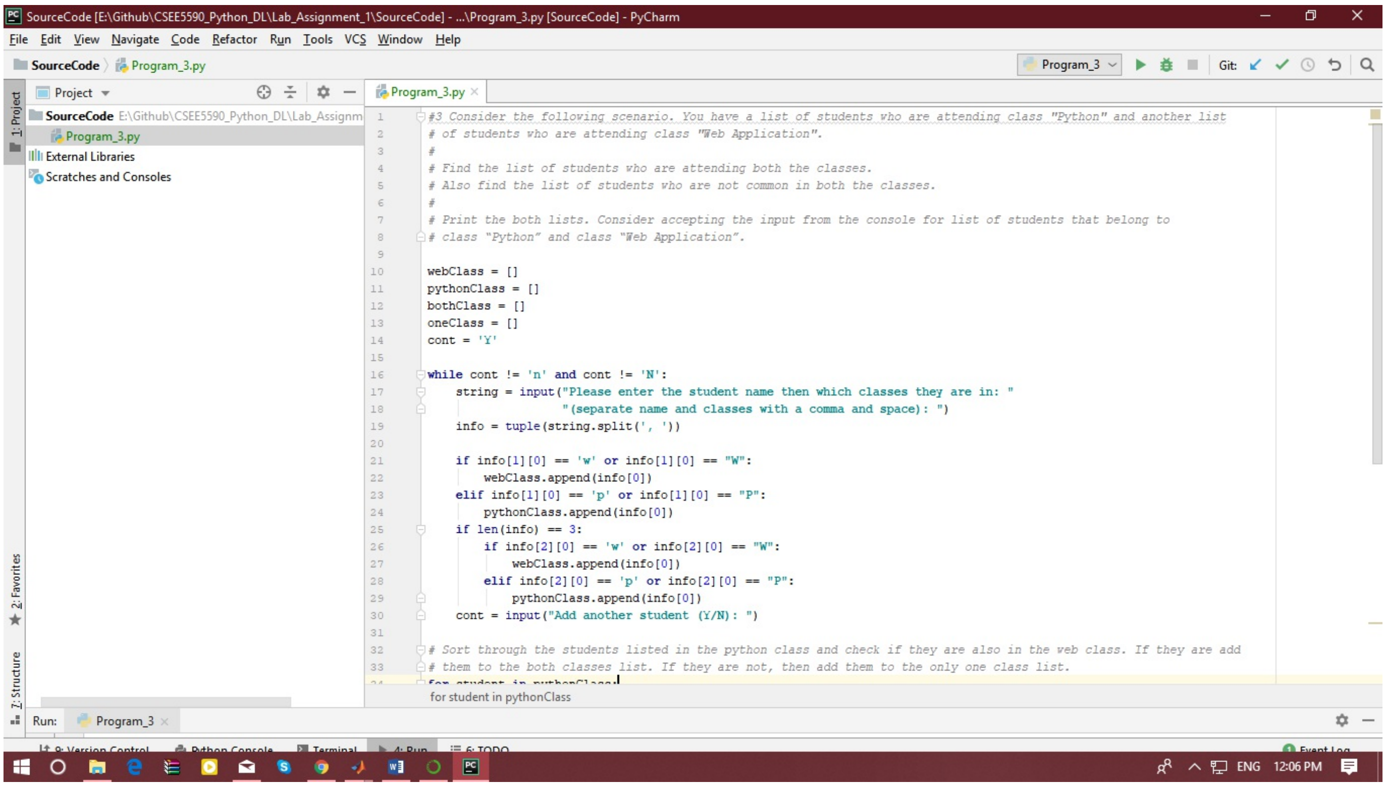
> ***Implementation***

The list of students who are attending the classes python and web can figured out by using sets(). To get the common students. In this, we take the input from the console which includes the list of python class students and web programming class students and then check both the lists and compare it to get the common students.

> ***Code Snippet:***

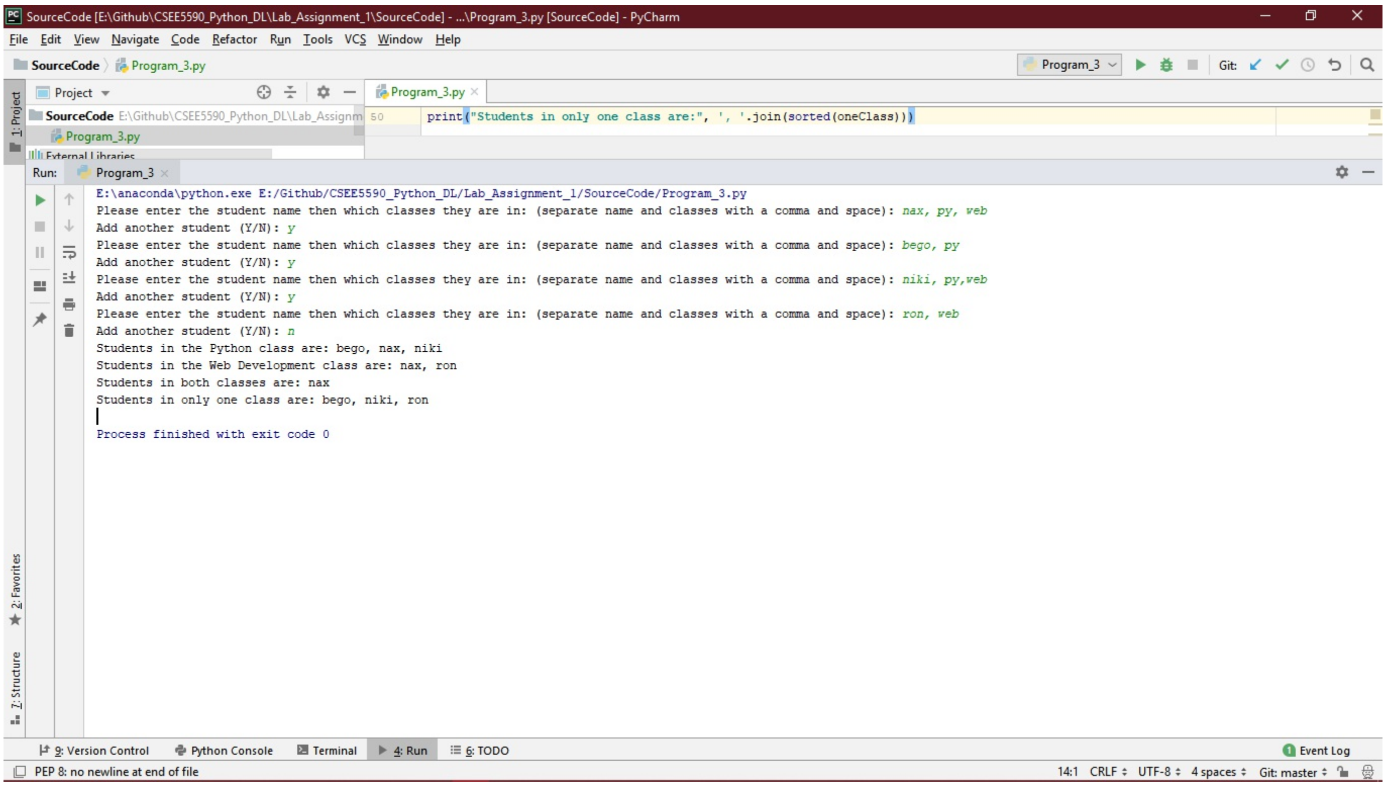


```
        webClass.append(info[0])
    elif info[1][0] == 'p' or info[1][0] == "P":
        pythonClass.append(info[0])
    if len(info) == 3:
        if info[2][0] == 'w' or info[2][0] == "W":
            webClass.append(info[0])
        elif info[2][0] == 'p' or info[2][0] == "P":
            pythonClass.append(info[0])
    cont = input("Add another student (Y/N): ")

# Sort through the students listed in the python class and check if they are also in the web class. If they are add
# them to the both classes list. If they are not, then add them to the only one class list.
for student in pythonClass:
    if student in webClass:
        bothClass.append(student)
    else:
        oneClass.append(student)

# Sort through the web class list and see if they are in the python class. If they are in the python class, then they
# would have already been added to the both classes list. Just those students only in the web class need to be added
# to one class list
for student in webClass:
    if student not in pythonClass:
        oneClass.append(student)

print("Students in the Python class are:", ', '.join(sorted(pythonClass)))
print("Students in the Web Development class are:", ', '.join(sorted(webClass)))
print("Students in both classes are:", ', '.join(sorted(bothClass)))
print("Students in only one class are:", ', '.join(sorted(oneClass)))
```



```python
#3 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 both the classes.
# Also find the list of students who are not common in both the classes.
#
# Print the both lists. Consider accepting the input from the console for list of students that belong to
# class “Python” and class “Web Application”.

webClass = []
pythonClass = []
bothClass = []
oneClass = []
cont = 'Y'

while cont != 'n' and cont != 'N':
    string = input("Please enter the student name then which classes they are in: "
                   "(separate name and classes with a comma and space): ")
    info = tuple(string.split(', '))

    if info[1][0] == 'w' or info[1][0] == "W":
        webClass.append(info[0])
    elif info[1][0] == 'p' or info[1][0] == "P":
        pythonClass.append(info[0])
    if len(info) == 3:
        if info[2][0] == 'w' or info[2][0] == "W":
            webClass.append(info[0])
        elif info[2][0] == 'p' or info[2][0] == "P":
            pythonClass.append(info[0])
    cont = input("Add another student (Y/N): ")

# Sort through the students listed in the python class and check if they are also in the web class. If they are add
# them to the both classes list. If they are not, then add them to the only one class list.
```

Input and Output



```python
print("Students in only one class are:", ', '.join(sorted(oneClass)))
```

```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Program_3.py
Please enter the student name then which classes they are in: (separate name and classes with a comma and space): nax, py, web
Add another student (Y/N): y
Please enter the student name then which classes they are in: (separate name and classes with a comma and space): bego, py
Add another student (Y/N): y
Please enter the student name then which classes they are in: (separate name and classes with a comma and space): niki, py,web
Add another student (Y/N): y
Please enter the student name then which classes they are in: (separate name and classes with a comma and space): ron, web
Add another student (Y/N): n
Students in the Python class are: bego, nax, niki
Students in the Web Development class are: nax, ron
Students in both classes are: nax
Students in only one class are: bego, niki, ron

Process finished with exit code 0
```

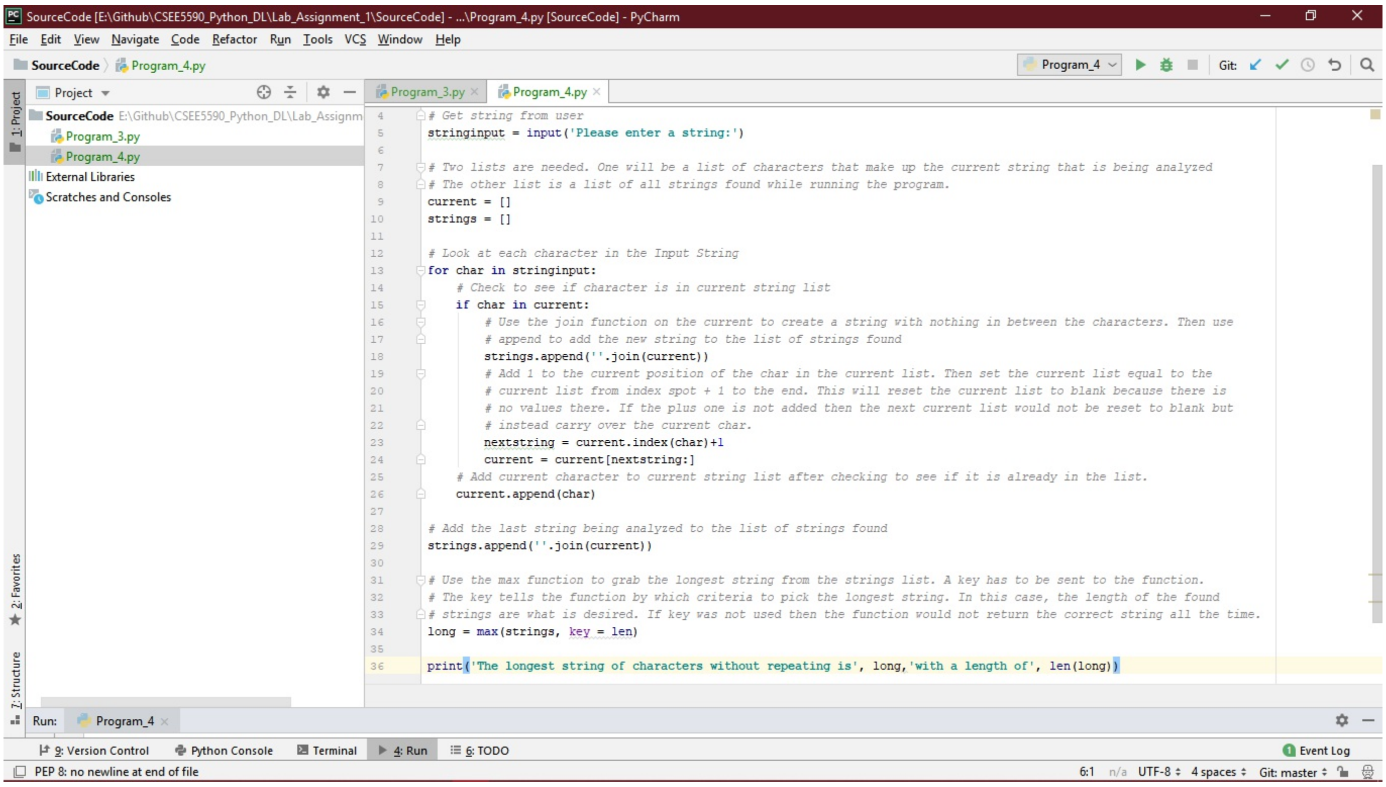
Question-4: Given a string, find the longest substring without repeating characters along with the length.

Input: “pwwkew”

Output: wke,3

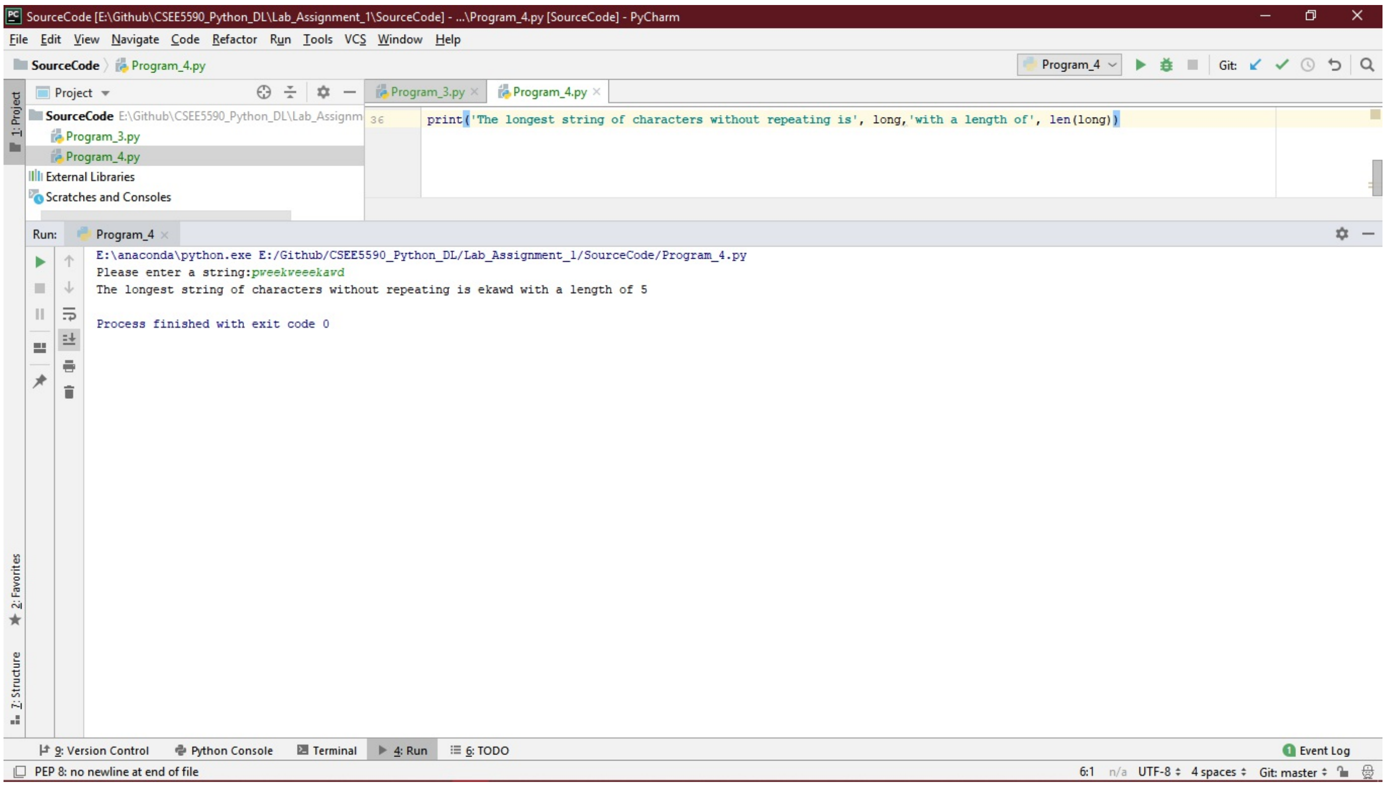
Solution-4:

Code Snippet:



```python
# Get string from user
stringinput = input('Please enter a string:')

# Two lists are needed. One will be a list of characters that make up the current string that is being analyzed
# The other list is a list of all strings found while running the program.
current = []
strings = []

# Look at each character in the Input String
for char in stringinput:
    # Check to see if character is in current string list
    if char in current:
        # Use the join function on the current to create a string with nothing in between the characters. Then use
        # append to add the new string to the list of strings found
        strings.append(''.join(current))
        # Add 1 to the current position of the char in the current list. Then set the current list equal to the
        # current list from index spot + 1 to the end. This will reset the current list to blank because there is
        # no values there. If the plus one is not added then the next current list would not be reset to blank but
        # instead carry over the current char.
        nextstring = current.index(char)+1
        current = current[nextstring:]
    # Add current character to current string list after checking to see if it is already in the list.
    current.append(char)

# Add the last string being analyzed to the list of strings found
strings.append(''.join(current))

# Use the max function to grab the longest string from the strings list. A key has to be sent to the function.
# The key tells the function by which criteria to pick the longest string. In this case, the length of the found
# strings are what is desired. If key was not used then the function would not return the correct string all the time.
long = max(strings, key = len)

print('The longest string of characters without repeating is', long,'with a length of', len(long))
```

Input and Output



```python
print('The longest string of characters without repeating is', long,'with a length of', len(long))
```

```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Program_4.py
Please enter a string:pweekweeekawd
The longest string of characters without repeating is ekawd with a length of 5

Process finished with exit code 0
```

Question-5:

Write a python program to create any one of the following management systems.

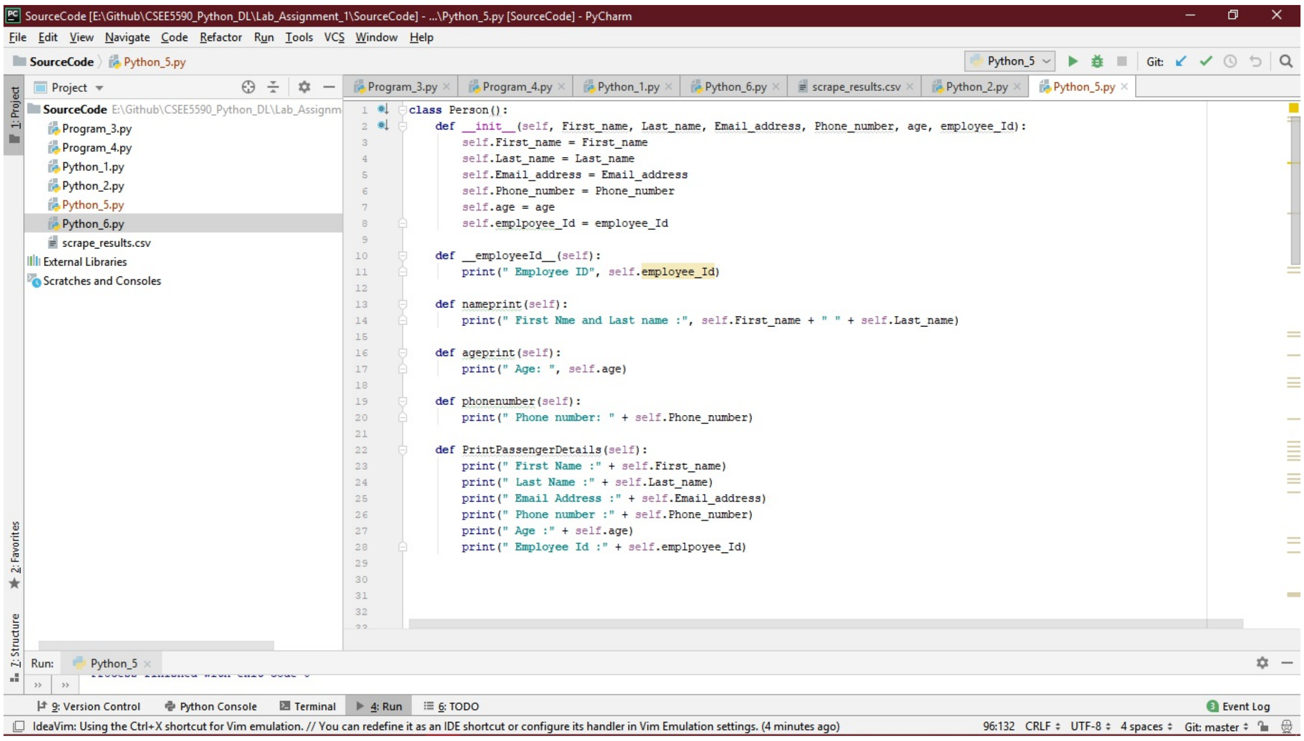
1. Airline Booking Reservation System (e.g. classes Flight, Person, Employee, Passenger etc.)

2. Library Management System(eg: Student, Book, Faculty, Department etc.)

Solution-5:

Airline Booking Reservation System:

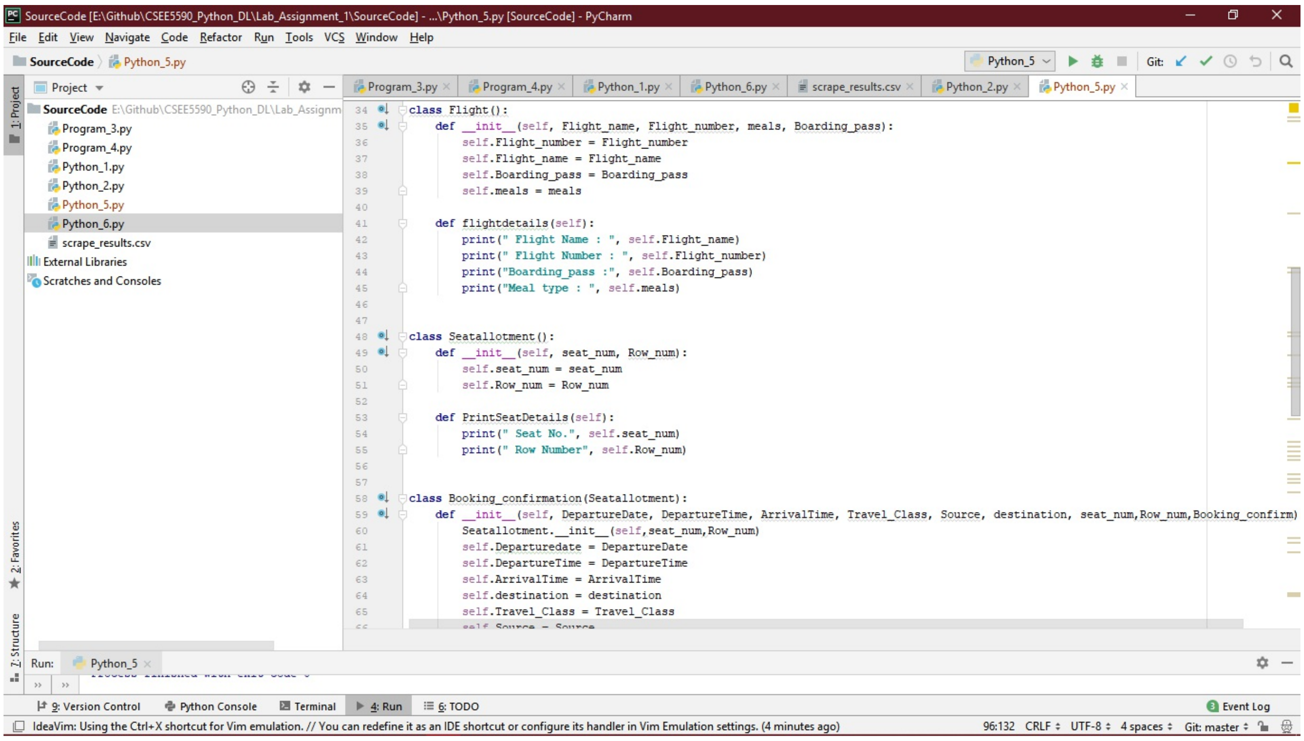
Code Snippet:



SourceCode [E:\Github\CSEE5590_Python_DL\Lab_Assignment_1\SourceCode] - ...\Python_5.py [SourceCode] - PyCharm

```python
class Person():
    def __init__(self, First_name, Last_name, Email_address, Phone_number, age, employee_Id):
        self.First_name = First_name
        self.Last_name = Last_name
        self.Email_address = Email_address
        self.Phone_number = Phone_number
        self.age = age
        self.emplpoyee_Id = employee_Id

    def __employeeId__(self):
        print(" Employee ID", self.employee_Id)

    def nameprint(self):
        print(" First Nme and Last name :", self.First_name + " " + self.Last_name)

    def ageprint(self):
        print(" Age: ", self.age)

    def phonenumber(self):
        print(" Phone number: " + self.Phone_number)

    def PrintPassengerDetails(self):
        print(" First Name :" + self.First_name)
        print(" Last Name :" + self.Last_name)
        print(" Email Address :" + self.Email_address)
        print(" Phone number :" + self.Phone_number)
        print(" Age :" + self.age)
        print(" Employee Id :" + self.emplpoyee_Id)
```

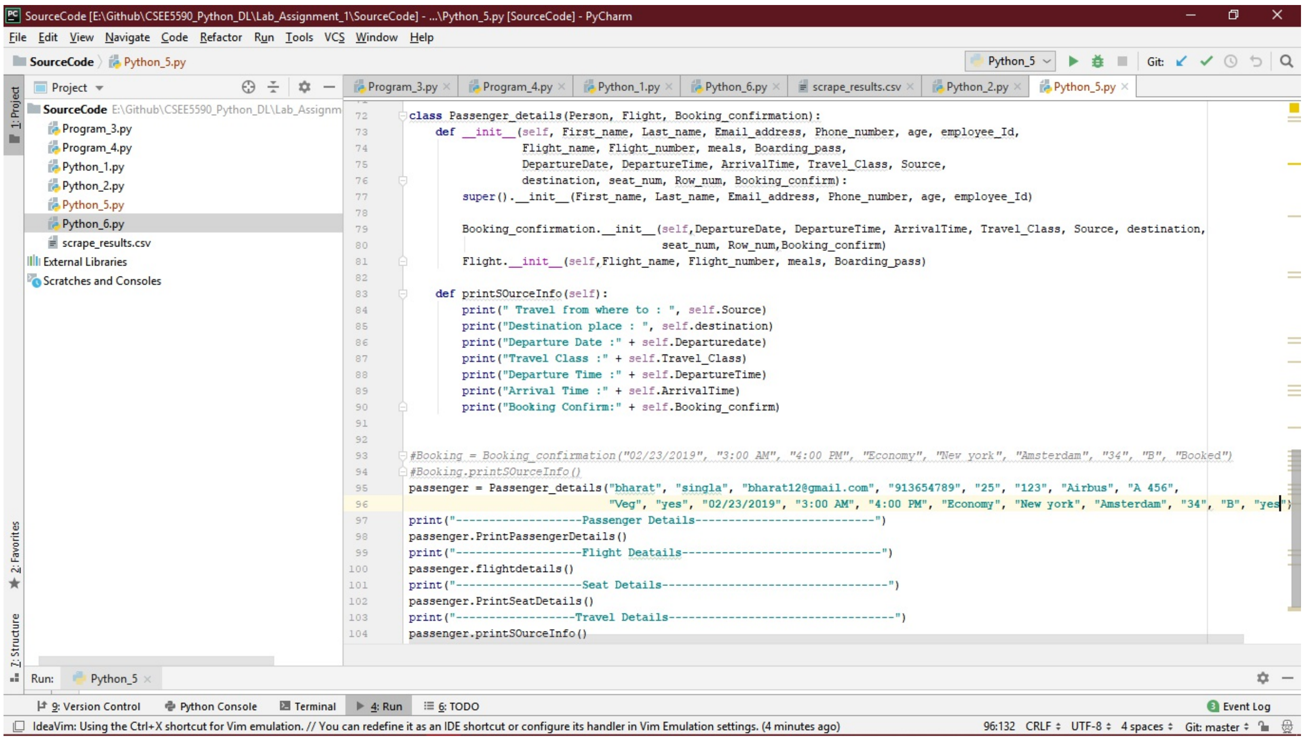
IdeaVim: Using the Ctrl+X shortcut for Vim emulation. // You can redefine it as an IDE shortcut or configure its handler in Vim Emulation settings. (4 minutes ago)



SourceCode [E:\Github\CSEE5590_Python_DL\Lab_Assignment_1\SourceCode] - ...\Python_5.py [SourceCode] - PyCharm

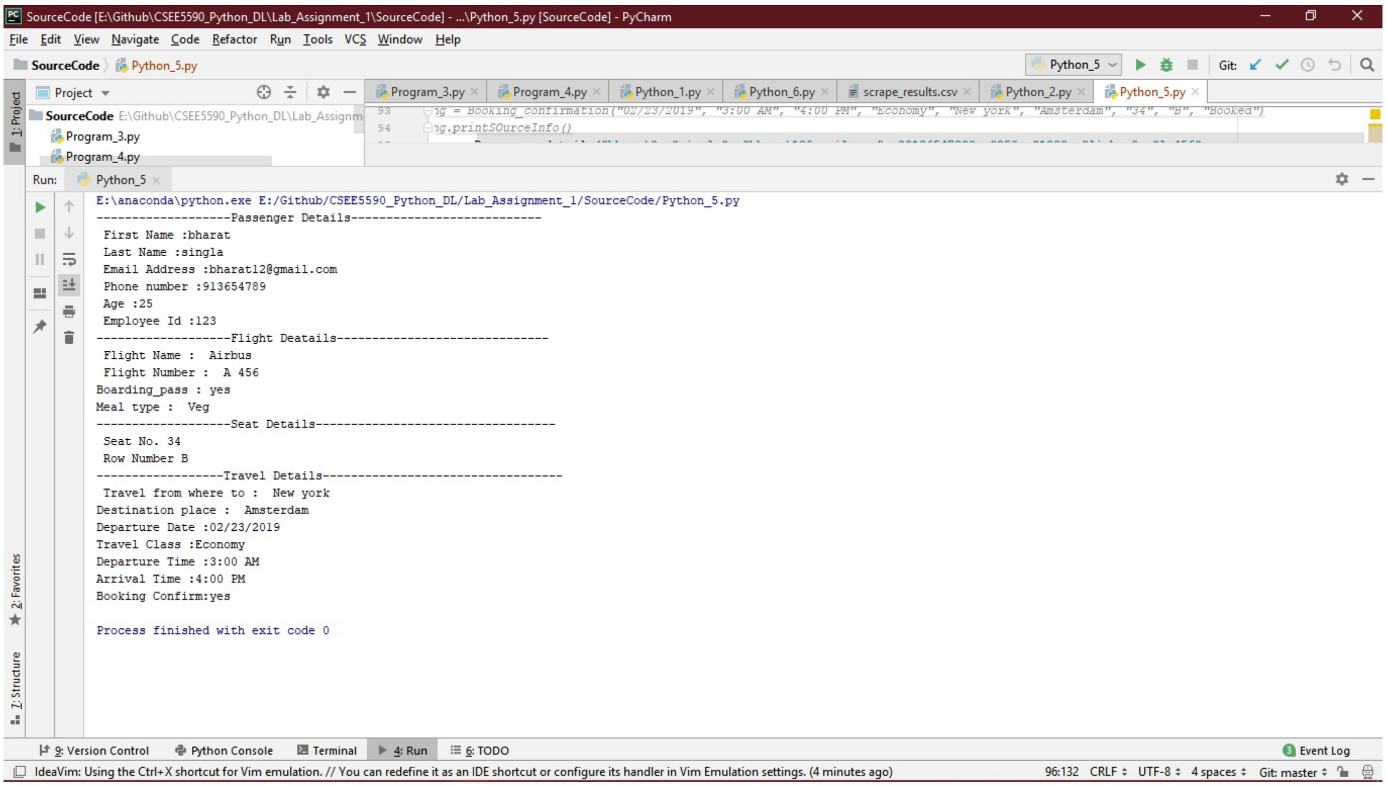
```python
class Flight():
    def __init__(self, Flight_name, Flight_number, meals, Boarding_pass):
        self.Flight_number = Flight_number
        self.Flight_name = Flight_name
        self.Boarding_pass = Boarding_pass
        self.meals = meals

    def flightdetails(self):
        print(" Flight Name : ", self.Flight_name)
        print(" Flight Number : ", self.Flight_number)
        print("Boarding_pass :", self.Boarding_pass)
        print("Meal type : ", self.meals)


class Seatallotment():
    def __init__(self, seat_num, Row_num):
        self.seat_num = seat_num
        self.Row_num = Row_num

    def PrintSeatDetails(self):
        print(" Seat No.", self.seat_num)
        print(" Row Number", self.Row_num)


class Booking_confirmation(Seatallotment):
    def __init__(self, DepartureDate, DepartureTime, ArrivalTime, Travel_Class, Source, destination, seat_num,Row_num,Booking_confirm)
        Seatallotment.__init__(self,seat_num,Row_num)
        self.Departuredate = DepartureDate
        self.DepartureTime = DepartureTime
        self.ArrivalTime = ArrivalTime
        self.destination = destination
        self.Travel_Class = Travel_Class
        self.Source = Source
```

IdeaVim: Using the Ctrl+X shortcut for Vim emulation. // You can redefine it as an IDE shortcut or configure its handler in Vim Emulation settings. (4 minutes ago)



```python
class Passenger_details(Person, Flight, Booking_confirmation):
    def __init__(self, First_name, Last_name, Email_address, Phone_number, age, employee_Id,
                 Flight_name, Flight_number, meals, Boarding_pass,
                 DepartureDate, DepartureTime, ArrivalTime, Travel_Class, Source,
                 destination, seat_num, Row_num, Booking_confirm):
        super().__init__(First_name, Last_name, Email_address, Phone_number, age, employee_Id)

        Booking_confirmation.__init__(self,DepartureDate, DepartureTime, ArrivalTime, Travel_Class, Source, destination,
                                      seat_num, Row_num,Booking_confirm)
        Flight.__init__(self,Flight_name, Flight_number, meals, Boarding_pass)

    def printSOurceInfo(self):
        print(" Travel from where to : ", self.Source)
        print("Destination place : ", self.destination)
        print("Departure Date :" + self.Departuredate)
        print("Travel Class :" + self.Travel_Class)
        print("Departure Time :" + self.DepartureTime)
        print("Arrival Time :" + self.ArrivalTime)
        print("Booking Confirm:" + self.Booking_confirm)


#Booking = Booking_confirmation("02/23/2019", "3:00 AM", "4:00 PM", "Economy", "New york", "Amsterdam", "34", "B", "Booked")
#Booking.printSOurceInfo()
passenger = Passenger_details("bharat", "singla", "bharat12@gmail.com", "913654789", "25", "123", "Airbus", "A 456",
                              "Veg", "yes", "02/23/2019", "3:00 AM", "4:00 PM", "Economy", "New york", "Amsterdam", "34", "B", "yes")
print("------------------Passenger Details--------------------------")
passenger.PrintPassengerDetails()
print("------------------Flight Deatails-----------------------------")
passenger.flightdetails()
print("------------------Seat Details--------------------------------")
passenger.PrintSeatDetails()
print("-----------------Travel Details-------------------------------")
passenger.printSOurceInfo()
```

Input and Output:



```
E:\anaconda\python.exe E:/Github/CSEE5590_Python_DL/Lab_Assignment_1/SourceCode/Python_5.py
-------------------Passenger Details--------------------------
 First Name :bharat
 Last Name :singla
 Email Address :bharat12@gmail.com
 Phone number :913654789
 Age :25
 Employee Id :123
-------------------Flight Deatails-----------------------------
 Flight Name :  Airbus
 Flight Number :  A 456
Boarding_pass : yes
Meal type :  Veg
-------------------Seat Details---------------------------------
 Seat No. 34
 Row Number B
------------------Travel Details----------------------------------
 Travel from where to :  New york
Destination place :  Amsterdam
Departure Date :02/23/2019
Travel Class :Economy
Departure Time :3:00 AM
Arrival Time :4:00 PM
Booking Confirm:yes

Process finished with exit code 0
```

Question-6:

Program a code which download a webpage contains a table using

Request library, then parse the page using Beautiful soup library. You should save the information about the states and their capitals in a file.

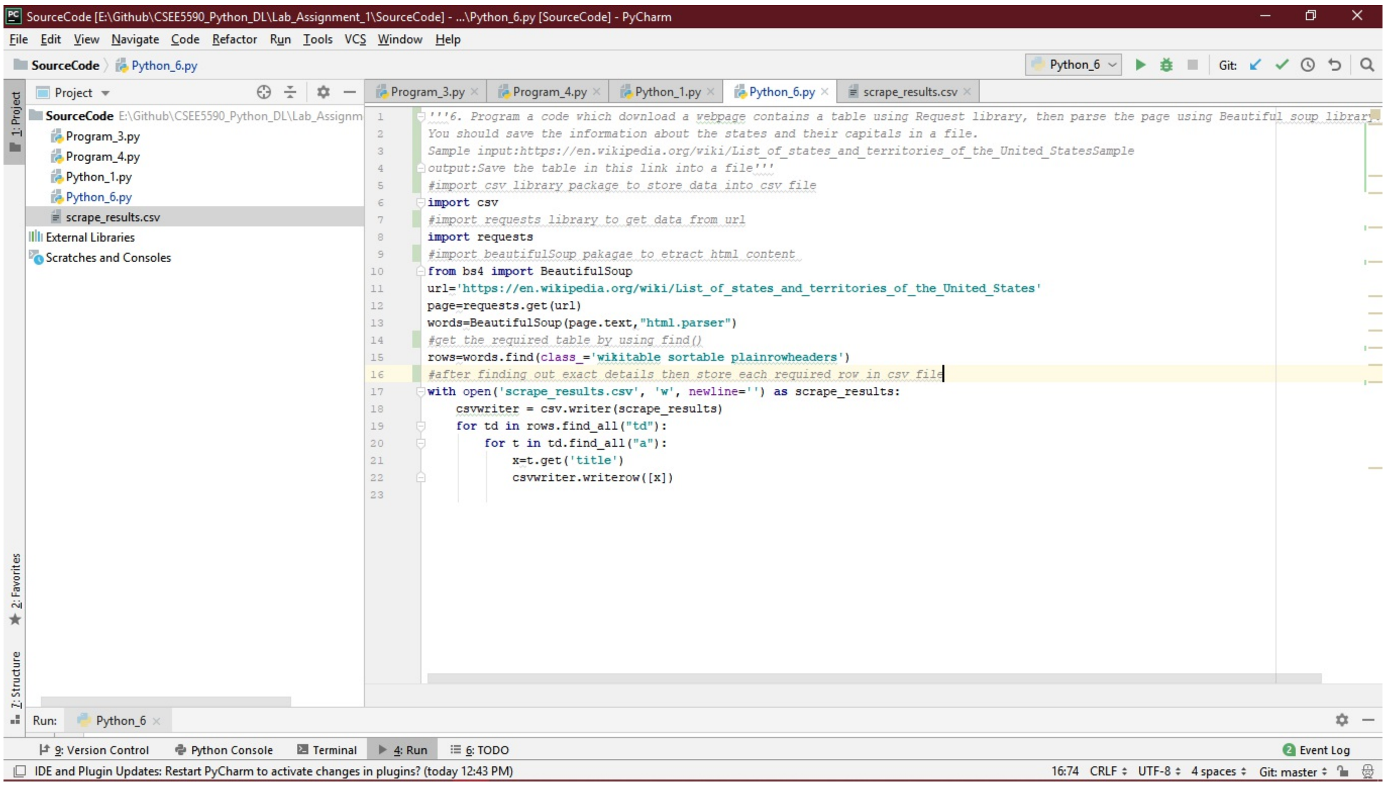
Input:https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_U

Output:Save the table in this link into a file

Solution-6:

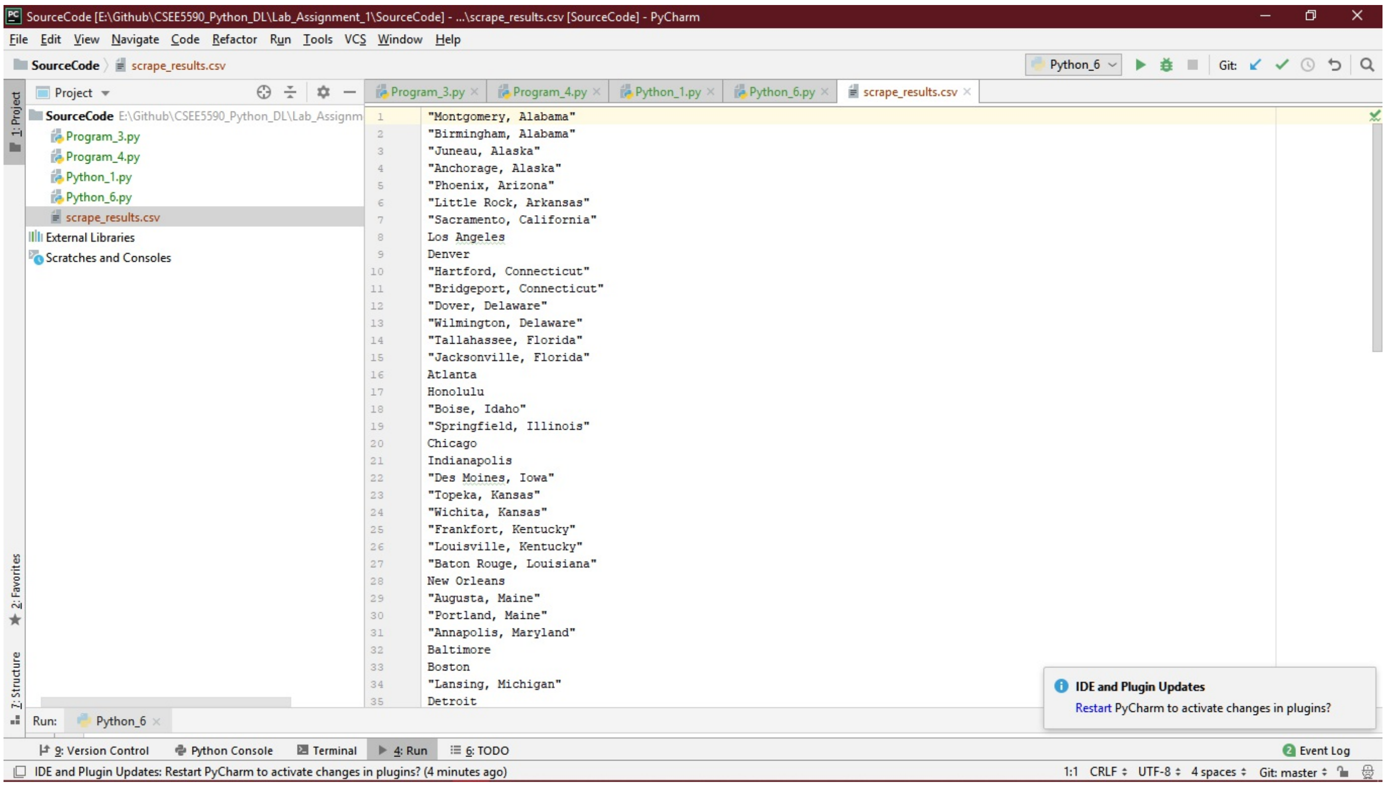
we extract list of cities and states in the file as output.

> ***Code Snippet:***


SourceCode [E:\Github\CSEE5590_Python_DL\Lab_Assignment_1\SourceCode] - ...\Python_6.py [SourceCode] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
SourceCode Python_6.py
Project
SourceCode E:\Github\CSEE5590_Python_DL\Lab_Assignm
Program_3.py
Program_4.py
Python_1.py
Python_6.py
scrape_results.csv
External Libraries
Scratches and Consoles
Program_3.py Program_4.py Python_1.py Python_6.py scrape_results.csv
Python_6
Git:
Run: Python_6
9: Version Control Python Console Terminal 4: Run 6: TODO
Event Log
IDE and Plugin Updates: Restart PyCharm to activate changes in plugins? (today 12:43 PM)
16:74 CRLF UTF-8 4 spaces Git: master

```python
'''6. Program a code which download a webpage contains a table using Request library, then parse the page using Beautiful soup library
You should save the information about the states and their capitals in a file.
Sample input:https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_StatesSample
output:Save the table in this link into a file'''
#import csv library package to store data into csv file
import csv
#import requests library to get data from url
import requests
#import beautifulSoup pakagae to etract html content
from bs4 import BeautifulSoup
url='https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States'
page=requests.get(url)
words=BeautifulSoup(page.text,"html.parser")
#get the required table by using find()
rows=words.find(class_='wikitable sortable plainrowheaders')
#after finding out exact details then store each required row in csv file
with open('scrape_results.csv', 'w', newline='') as scrape_results:
    csvwriter = csv.writer(scrape_results)
    for td in rows.find_all("td"):
        for t in td.find_all("a"):
            x=t.get('title')
            csvwriter.writerow([x])
```

> ***Input and Output***


SourceCode [E:\Github\CSEE5590_Python_DL\Lab_Assignment_1\SourceCode] - ...\scrape_results.csv [SourceCode] - PyCharm

```
"Montgomery, Alabama"
"Birmingham, Alabama"
"Juneau, Alaska"
"Anchorage, Alaska"
"Phoenix, Arizona"
"Little Rock, Arkansas"
"Sacramento, California"
Los Angeles
Denver
"Hartford, Connecticut"
"Bridgeport, Connecticut"
"Dover, Delaware"
"Wilmington, Delaware"
"Tallahassee, Florida"
"Jacksonville, Florida"
Atlanta
Honolulu
"Boise, Idaho"
"Springfield, Illinois"
Chicago
Indianapolis
"Des Moines, Iowa"
"Topeka, Kansas"
"Wichita, Kansas"
"Frankfort, Kentucky"
"Louisville, Kentucky"
"Baton Rouge, Louisiana"
New Orleans
"Augusta, Maine"
"Portland, Maine"
"Annapolis, Maryland"
Baltimore
Boston
"Lansing, Michigan"
Detroit
```

IDE and Plugin Updates
Restart PyCharm to activate changes in plugins?

IDE and Plugin Updates: Restart PyCharm to activate changes in plugins? (4 minutes ago)