G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Pune	
NAME:	Shreeraj Anand Raul
ENROLLMENT NO.:	20200000363
ROLL NO.:	60
SUBJECT:	Foundation of Data Analytics with Python
A.Y.:	2020-21
BATCH:	SY B.Tech IT
Github:	https://github.com/ShreerajRaul/ghriet/tree/main/semesterIII/FADP

Assignment 1

problem Defination:

write a python program to multiply 2 matrices.

Theory:

In this assignment, we will learn to multiply matrices using two different ways: nested loop and, nested list comprehesion.

To understand this example, you should have the knowledge of the following python programming topic:

python for loop
python List
python Matrices and Numpy Arrays
In python we can implement a matrix as nested list (list inside list).

We can treat each element as a row of the matrix

For example X=[[1,2], [4,5], [3,6]] would represent a 3X2 matrix.

The first row can be selected as X[0], and the element in first row, first column can be selected as X[0][0]

Multiplication of two matrix x and Y is defined only if the number of columns in x is equal to the number of rows Y.

If x is a n xm matrix and Y is a mx1 matrix then, xY is defined and has the diamension

nx1 (but YX is not defined). Here are a couple of ways to implement matrix multiplication in python.

Source code: Matrix multiplication using Nested loop.

```
# program to multiply two matrices using nested loops
# 3X3 matrix
X = [[12, 7, 3],
     [4, 5, 6],
     [7, 8, 9]]

# 3x4 matrix
Y = [[5, 8, 1, 2],
     [6, 7, 3, 0],
     [4, 5, 9, 1]]

# result is 3x4
result = [[0, 0, 0, 0],
          [0, 0, 0, 0],
          [0, 0, 0, 0]]
for i in range(len(X)):
    # iterate through columns of Y
    for j in range(len(Y[0])):
        # iterate through rows of Y
        for k in range(len(Y)):
            result[i][j] += X[i][k] * Y[k][j]
for r in result:
print(r)
```

Output:

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

In this program we have use nested for loops to iterate through each row and each column. we accumulate the sum of product in the result

This techniques is simple but computationally expensive as we increase the order of the matrix.

For larger matrix operations we recommend optimized software packages like Numpy which is several (in the order of 1000) times faster than the above code

Source Code: Matrix Multiplication using Nested List Comprehension

```
# Program to multiply two matrix using list comprehension

# 3X3 matrix
X = [[12, 7, 3],
     [4, 5, 6],
     [7, 8, 9]]

# 3X4 Matrix
Y = [[5, 8, 1, 2],
     [6, 7, 3, 0],
     [4, 5, 9, 1]]
```

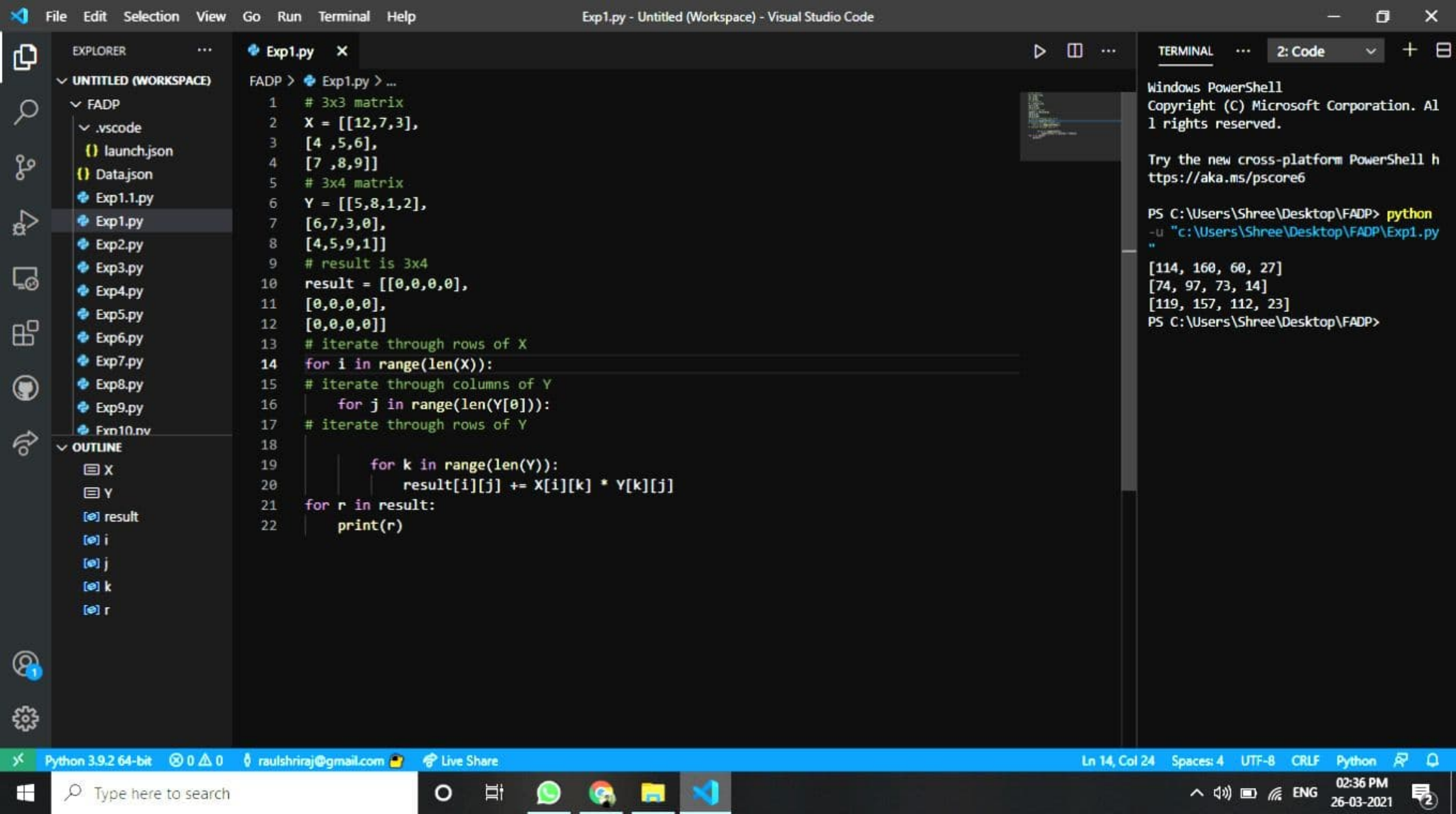
result is 3X4

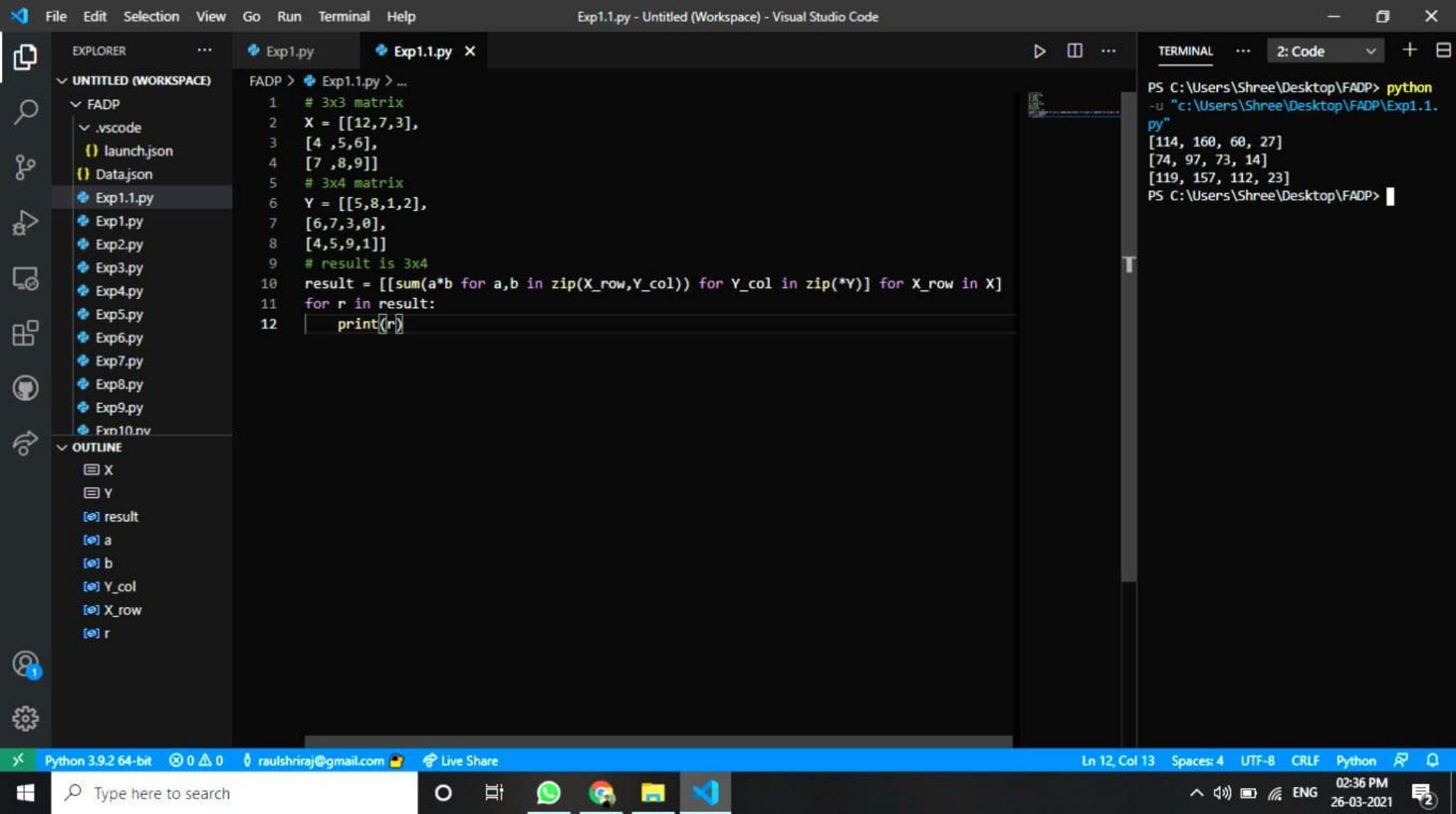
```
result = [[sum(a*b for a,b in zip(X_row, Y_col)) for Y_col in zip(*Y)] for X_row in X]

for r in result:
print(r)
```

The output of this program is the same as above. To understand the above code we must first know about built-in function zip() and unpacking argument list using * operator.

We have used nested list comprehension to iterate through each element in the matrix. The code looks complicated and unreadable first But once you get the hang of list comprehensions, you will probably not go back to nested loops.

Conclusion:
In this way we have implemented python program to multiply 2 matrices


File Edit Selection View Go Run Terminal Help
Exp1.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
X
Y
result
i
j
k
r
Exp1.py
FADP > Exp1.py > ...
1 # 3x3 matrix
2 X = [[12,7,3],
3 [4 ,5,6],
4 [7 ,8,9]]
5 # 3x4 matrix
6 Y = [[5,8,1,2],
7 [6,7,3,0],
8 [4,5,9,1]]
9 # result is 3x4
10 result = [[0,0,0,0],
11 [0,0,0,0],
12 [0,0,0,0]]
13 # iterate through rows of X
14 for i in range(len(X)):
15 # iterate through columns of Y
16 for j in range(len(Y[0])):
17 # iterate through rows of Y
18
19 for k in range(len(Y)):
20 result[i][j] += X[i][k] * Y[k][j]
21 for r in result:
22 print(r)
TERMINAL
2: Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp1.py"
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
raulshriraj@gmail.com
Live Share
Ln 14, Col 24
Spaces: 4
UTF-8
CRLF
Python
Type here to search
ENG
02:36 PM
26-03-2021


File Edit Selection View Go Run Terminal Help
Exp1.1.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
X
Y
result
a
b
Y_col
X_row
r
Exp1.py
Exp1.1.py
FADP > Exp1.1.py > ...
1 # 3x3 matrix
2 X = [[12,7,3],
3 [4 ,5,6],
4 [7 ,8,9]]
5 # 3x4 matrix
6 Y = [[5,8,1,2],
7 [6,7,3,0],
8 [4,5,9,1]]
9 # result is 3x4
10 result = [[sum(a*b for a,b in zip(X_row,Y_col)) for Y_col in zip(*Y)] for X_row in X]
11 for r in result:
12 print(r)
TERMINAL
2: Code
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp1.1.py"
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
raulshriraj@gmail.com
Live Share
Ln 12, Col 13
Spaces: 4
UTF-8
CRLF
Python
Type here to search
ENG
02:36 PM
26-03-2021

Assignment 2

Problem Definition:
writen a python program to find a root of given quadratic equation.

Theory:
This program computes roots of a quadratic equation when coefficients a, b and c are known.

To understand this example you should have the knowledge of this following Python programming topics:

Python Data Types
Python Input, Output and Import
python Operators

The standard form of a quadratic equation is:

$ax^2+bx+c=0$, where
a, b and c are real numbers and
$a \neq 0$

Source Code:

```
# Solve the quadratic equation ax**2+bx+c=0
# import complex math module
import cmath
a=1
b=5
c=6

# calculate the discriminant
d = (b**2) - (4*a*c)
```

Shuf

```
# find two solutions

sol1 = (-b-cmath.sqrt(d))/(2*a)

sol2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(sol1,
sol2))
```

Output:

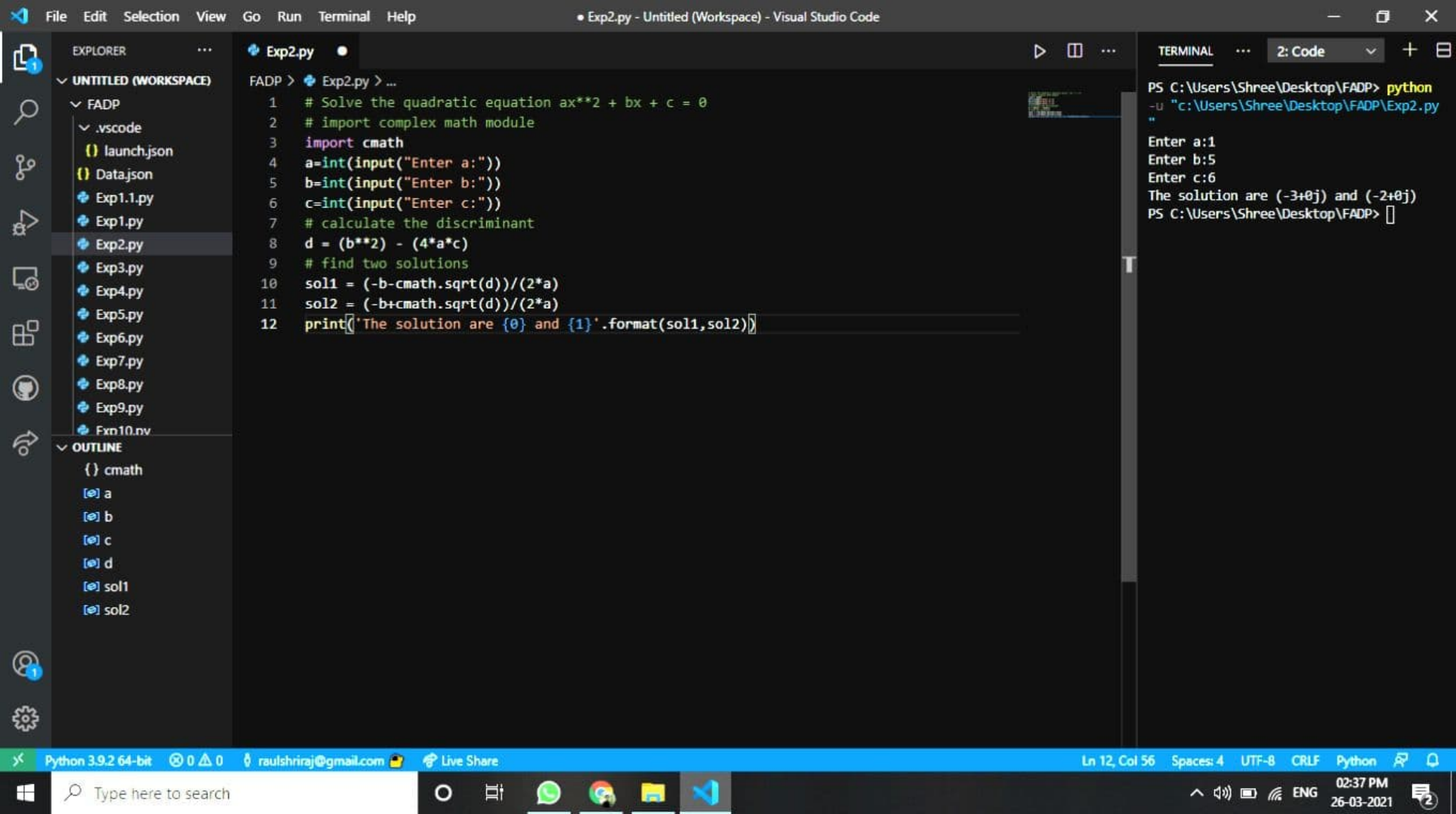
```
Enter a: 1
Enter b: 5
Enter c: 6
```

The solution are (-3+0j) and (-2+0j)
we have imported the cmath module to perform complex square root. first we calculate the discriminant and then find the two solutions of the quadratic equation.

you can change the value of a, b and c in the above program and test this program.

Conclusion:

In this way have implemented python program to find a root of given quadratic equation


File Edit Selection View Go Run Terminal Help
• Exp2.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
cmath
a
b
c
d
sol1
sol2
Exp2.py
FADP > Exp2.py > ...
1 # Solve the quadratic equation ax**2 + bx + c = 0
2 # import complex math module
3 import cmath
4 a=int(input("Enter a:"))
5 b=int(input("Enter b:"))
6 c=int(input("Enter c:"))
7 # calculate the discriminant
8 d = (b**2) - (4*a*c)
9 # find two solutions
10 sol1 = (-b-cmath.sqrt(d))/(2*a)
11 sol2 = (-b+cmath.sqrt(d))/(2*a)
12 print('The solution are {0} and {1}'.format(sol1,sol2))
TERMINAL
2: Code
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp2.py"
Enter a:1
Enter b:5
Enter c:6
The solution are (-3+0j) and (-2+0j)
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
raulshriraj@gmail.com
Live Share
Ln 12, Col 56
Spaces: 4
UTF-8
CRLF
Python
Type here to search
ENG
02:37 PM
26-03-2021

Assignment 3

Date
Page 7

Problem Definition:

Import given csv file in python and perform following operation on it

a) Reading specific Rows
b) Reading specific columns

Theory:

This program explains how to load and parse a ~~CSV~~ csv file ~~the~~ python.

CVs (comma-separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular-data (numbers and text) in plain text. Each line of the file is a data record. Each record consits of one or more fields, separated by commas. The use of the comma as a field separater is the name for this file format.

For working CSV files in python, there is an inbuilt module called CSV.

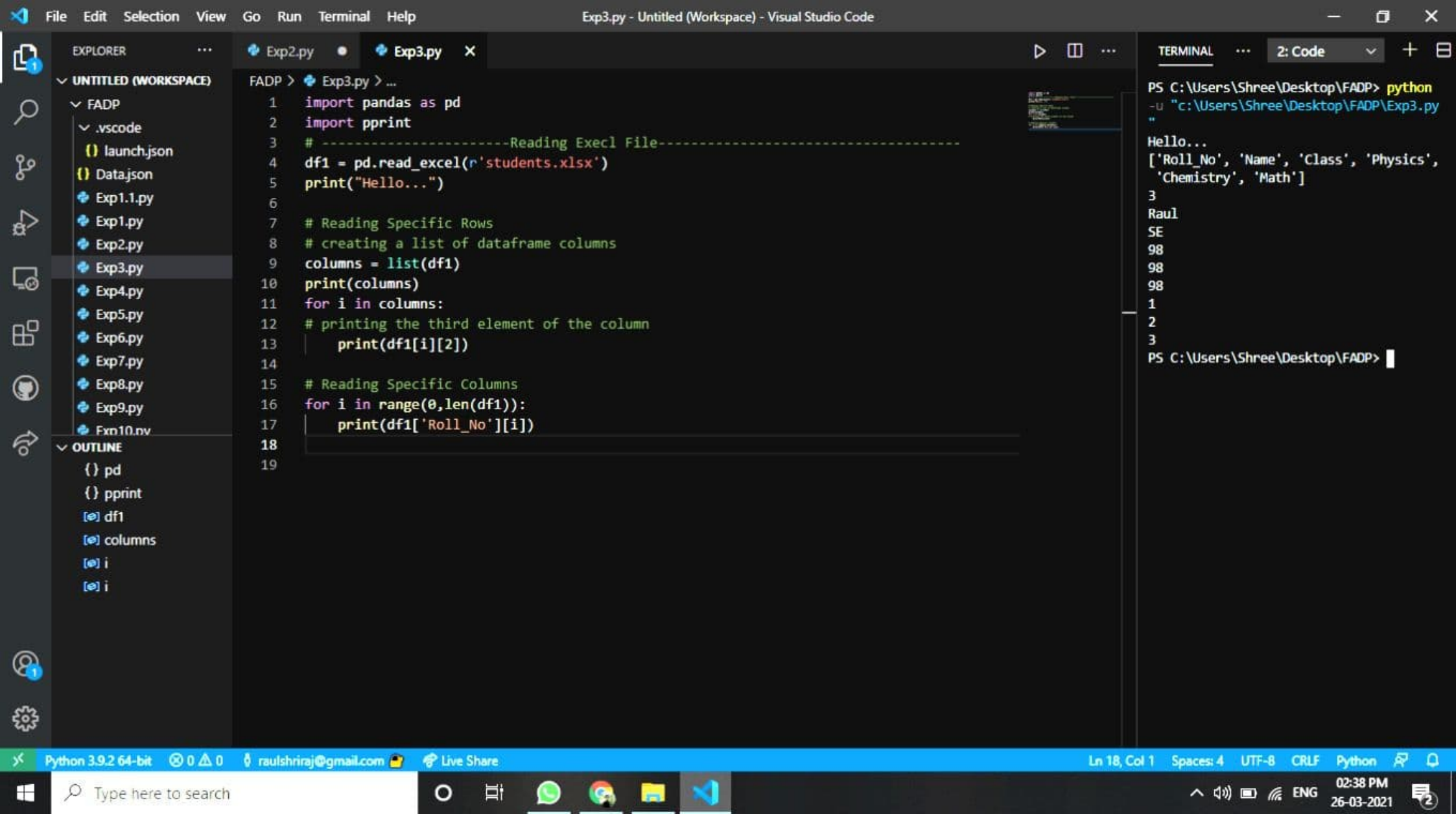
Code:

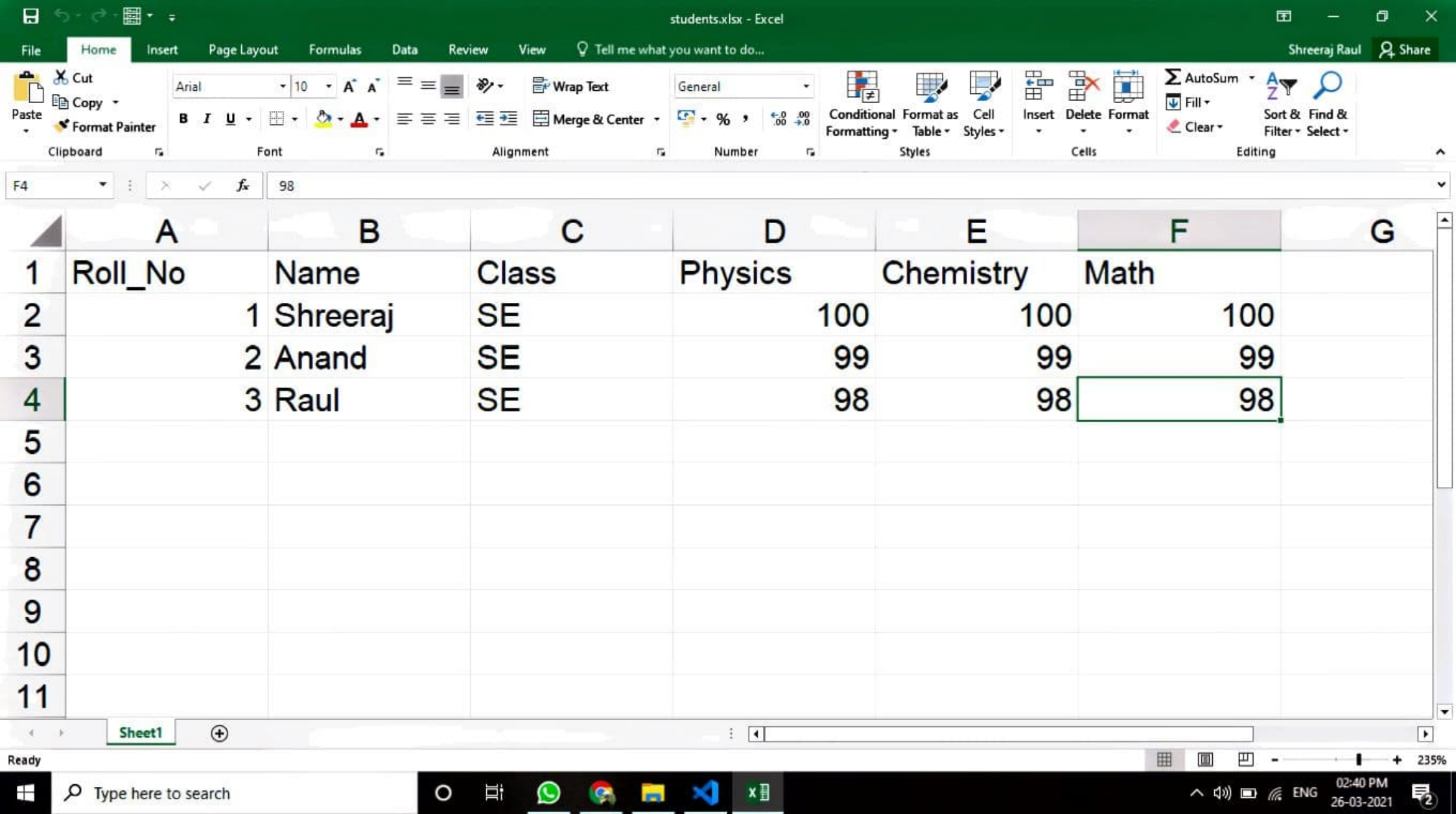
```
import pandas as pd
import pprint

#----Reading Excel file----
df1= pd.read_excel('C:/Users/XYZ/Desktop/python_file/students_data.xlsx', encoding ='utf-8')
print("Hello....")
```

```
# Reading specific Rows
# creating a list of dataframe columns
columns = list(df1)
print(df[i][2] columns)
for i in columns:
    # printing the third element of the columns
    print(df1[i][2])

# Reading Specific Columns.
for i in range(0, len(df1)):
    print(df['Roll-No'][i])
```

Conclusion:

In this way. have implement python program to Import given csv file in program and perform the operations on it.


File Edit Selection View Go Run Terminal Help
Exp3.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
pd
pprint
df1
columns
i
i
Exp2.py
Exp3.py
FADP > Exp3.py > ...
1 import pandas as pd
2 import pprint
3 # ----------Reading Execl File----------
4 df1 = pd.read_excel(r'students.xlsx')
5 print("Hello...")
6
7 # Reading Specific Rows
8 # creating a list of dataframe columns
9 columns = list(df1)
10 print(columns)
11 for i in columns:
12 # printing the third element of the column
13 print(df1[i][2])
14
15 # Reading Specific Columns
16 for i in range(0,len(df1)):
17 print(df1['Roll_No'][i])
18
19
TERMINAL
2: Code
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp3.py"
Hello...
['Roll_No', 'Name', 'Class', 'Physics', 'Chemistry', 'Math']
3
Raul
SE
98
98
98
1
2
3
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
0 0
raulshriraj@gmail.com
Live Share
Ln 18, Col 1
Spaces: 4
UTF-8
CRLF
Python
Type here to search
ENG
02:38 PM
26-03-2021


students.xlsx - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Shreeraj Raul
Share
Paste
Cut
Copy
Format Painter
Clipboard
Arial
10
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
F4
98
Roll_No
Name
Class
Physics
Chemistry
Math
1 Shreeraj SE 100 100 100
2 Anand SE 99 99 99
3 Raul SE 98 98 98
Sheet1
Ready
235%
Type here to search
ENG
02:40 PM
26-03-2021

Assignment 4

Date
Page 9

Problem Definition:

Perform following operations assignment no 3
a) Redding specific Columns and Rows
b) Reading specific Columns for a Range of Rows

Theory:

Interation is a general term for taking each item of something, one after another. Pandas Dataframe consists of rows and columns so, in order to iterate over dataframe. we have to iterate a data frame like a dictionary. In a dictionary, we interate over the keys of the object in the same way we have to iterate in dataframe

In Pandas Dataframe we can iterate an element in two ways:

Iterating over rows
Iterating over columns.
Iterating over rows:
In order to iterate over rows, we can use three function interitems(), iterrows(), itertuples(). These three function will help in iteration over rows.
Interation over rows using interrows()
Inorder to iterate over rows, we apply a iterrows() function this function return each index value along with a series containing thd data in each row.

Code:

```
Import pandas as pd
import pprint
# ----. Reading Exced file- - - -
df1= pd.read_excel ('c:/users/XYZ/Desktop/Python-
    file/Students_data.xlsx', encoding='utf-8')
 print("Hello..!")
 Students_data=[]
for i in range(0, len(df1)):
 marks={}
 marks['Roll-No']=df1['Roll-No'][i]
 Marks['Name']=df1['Name'][i]
 Marks['physics']=df1['Physics'][i]
 Marks['chemistry']=df1['chemistry'][i]
 Marks['Math']=df1['Math'][i]

 Student_data.append(marks)
 # print(student_data)
 #Pprint.pprint(Student_data)
 # Reading Specific columns to range of rows
 print(df['Name'][2:5])
```

Conclusion:

In this way have implemented python program to Import given CSV file in program and perform the operations on it

File Edit Selection View Go Run Terminal Help — Exp4.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- FADP
  - .vscode
    - launch.json
  - ~$students.xlsx
  - Data.json
  - Exp1.1.py
  - Exp1.py
  - Exp2.py
  - Exp3.py
  - Exp4.py
  - Exp5.py
  - Exp6.py
  - Exp7.py
  - Exp8.py
  - Exp9.py

OUTLINE

- pd
- pprint
- df1
- students_data
- i
- marks

Exp4.py

FADP > Exp4.py > ...

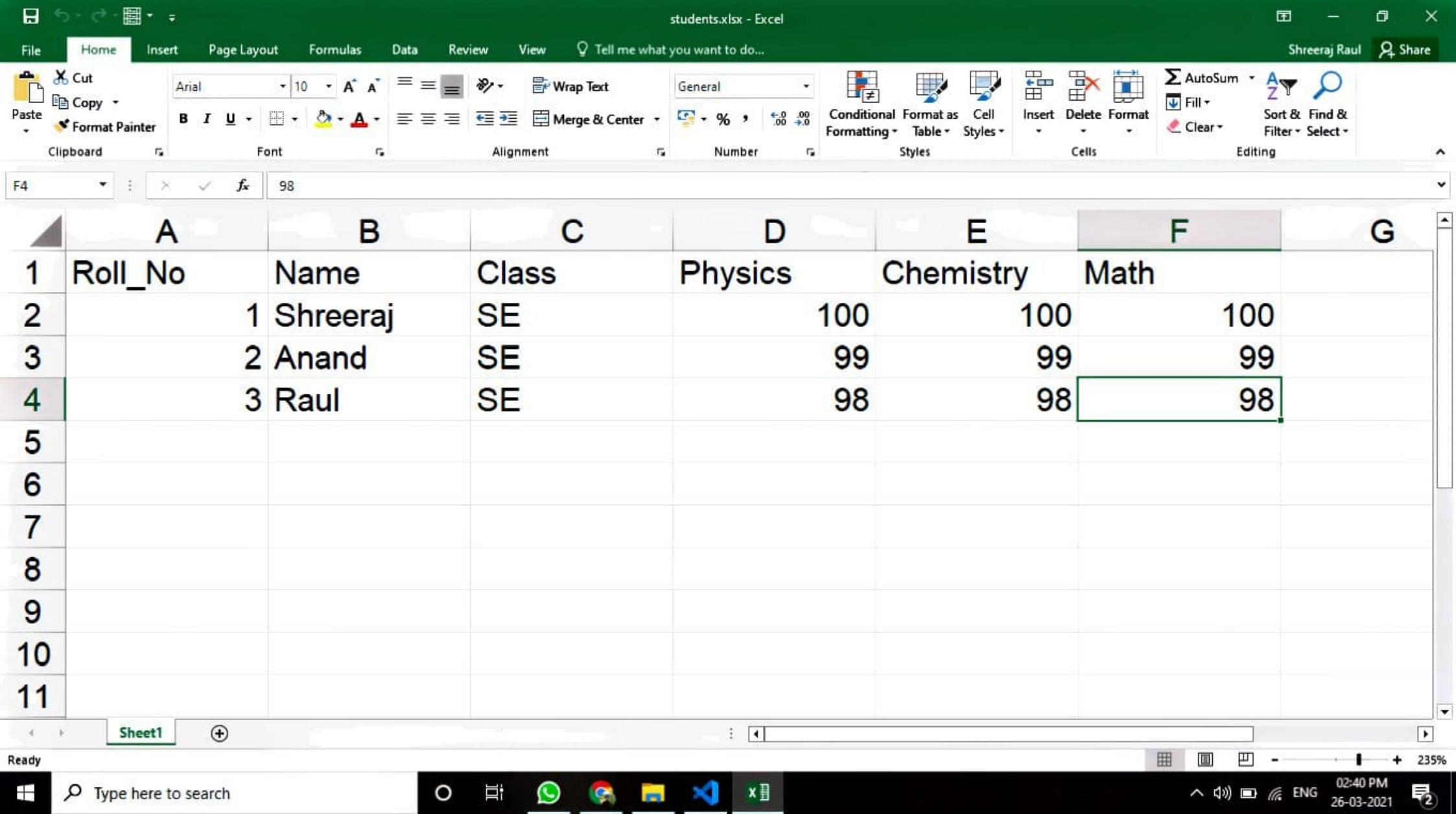
```python
import pandas as pd
import pprint
# ----------------------Reading Execl File-----------------------------------
df1 = pd.read_excel(r'students.xlsx')
print("Hello...")
students_data = []
for i in range(0,len(df1)):
    marks = {}
    marks['Roll_No'] = df1['Roll_No'][i]
    marks['Name'] = df1['Name'][i]
    marks['Physics'] = df1['Physics'][i]
    marks['Chemistry'] = df1['Chemistry'][i]
    marks['Math'] = df1['Math'][i]

students_data.append(marks)
#print(students_data)
#pprint.pprint(students_data)
# Reading specific columns to range of rows
print(df1['Name'][2:5])
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp4.py"
Hello...
2    Raul
Name: Name, dtype: object
PS C:\Users\Shree\Desktop\FADP>
```

Python 3.9.2 64-bit | 0 0 | raulshriraj@gmail.com | Live Share | Ln 4, Col 23 | Spaces: 4 | UTF-8 | CRLF | Python

Type here to search | ENG | 02:40 PM 26-03-2021


students.xlsx - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Shreeraj Raul
Share
Paste
Cut
Copy
Format Painter
Clipboard
Arial
10
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
F4
98
Roll_No
Name
Class
Physics
Chemistry
Math
1 Shreeraj SE 100 100 100
2 Anand SE 99 99 99
3 Raul SE 98 98 98
Sheet1
Ready
235%
Type here to search
ENG
02:40 PM
26-03-2021

Assignment: 5

Date
Page 11

Problem Definition:
Import given JSON data file in to python program and perform following operations on it
a) Redding JSON file as Records.

Theory:

The full-form of JSON in Javal Script Object Notation. It means that a script (excutable) file which is made of text in a programming language, is used store and transfer the data. python supports JSON through a built-in packa-ge called json. To use this feature, we import the json package in python script. The text in JSON is done through quoted string which contains the value in key-value mapping within {}

Reading From JSON

It's pretty easy to load a JSON object in python python has a built-in package called json, which can be used to work with JSON data, It's done by using the JSON module, which provides us with a lot of method which among loads() and load() methods are gonna help us to read the JSON file.

Deserialization of JSON

The Deserialization of JSON means the con-Version of JSON objects into their respective python object. The loads() / load() method is used for it. If you have used JSON data from anoth-er program or obtained as a string format of

Sfuj

JSON, then it can easily be deserialized with loads() /load(), which is usually used to load from string, otherwise the root object is in list or dict. see the following table given below:

JSON OBJECT	PYTHON OBJECT
object	dict
array	list
string	str
null	None
number (int)	int
number (real)	float
true	True
false	False

json.load(): json.load() accepts file object, parses the JSON data, populates a python dictionary with the data and returns it back to you.

Syntax:

json.load(file object)

Example: suppose the JSON file looks like this:

```
{ "emp_details": [
    {
      "emp_name": "shubham",
      "email": "shubham@gmail.com",
      "Job_Profile": "Intern"
    },
```

```
{
  "emp-name": "shreeraj",
  "email" : "shreeraj@gmail.com",
  "Job-Profile": "developer"
},
{
  "emp-name": "XYZ"
  "email": "xyz@gmail.com"
  "Job-profile": "full time"
}
]
}
```

we want to read the content of this file. Below is the implementation.

```
# python program to read JSON file

import json
# opening JSON file

f= open('data.json',)

# returns JSON objects as
# a dictionary
data = json.load(f)

# Iterating through the JSON
# list
for i in data['emp_details']:
print(i)
```

Ans

```
# closing file
f.close()
```

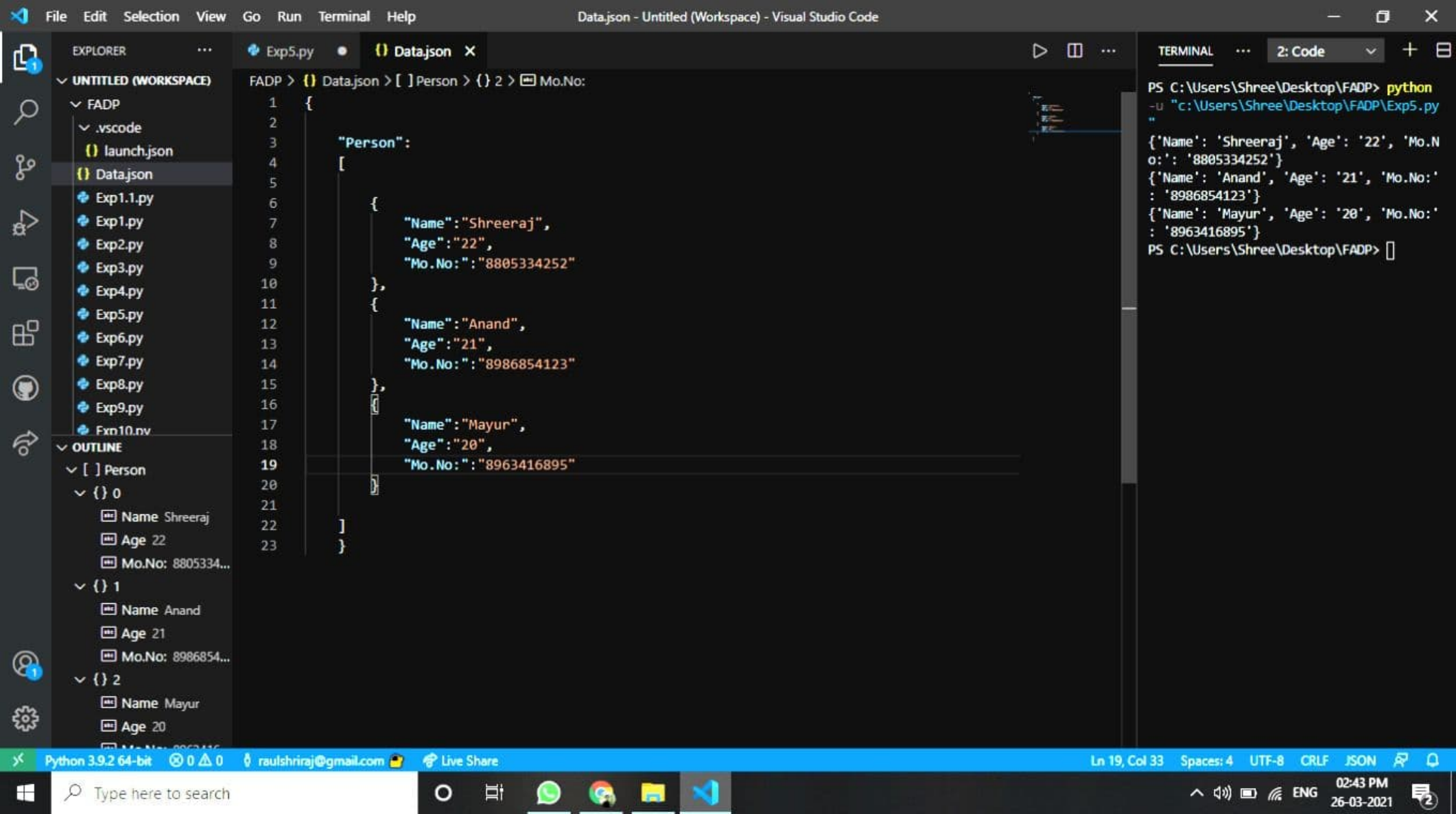
output:

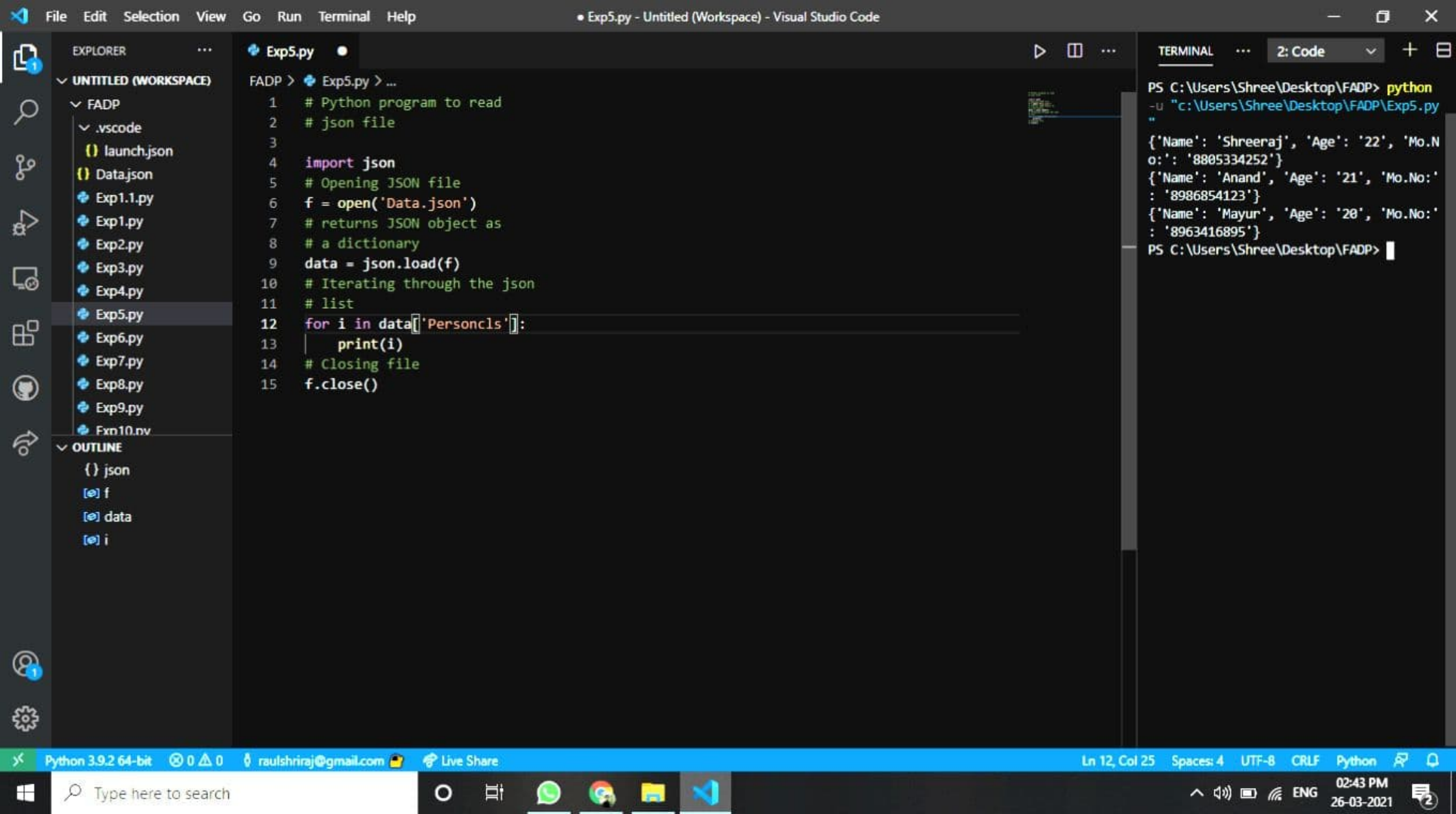
```
{'emp_name': 'shubham', 'email': 'shubham@gmail.com', 'Job-Profile': 'intern'}

{'emp_name': 'shreeraj', 'email': 'shreeraj@gmail.com', 'Job_Profile': 'developer'}

{'emp_name': 'xyz', 'email': 'xyz@gmail.com', 'Job-Profile': 'full time'}
```

Conclusion:

Hence we have implemented python program to Import given JSON data file in to python program and perform following Operations on it

a) Reading JSON file as Records.


File Edit Selection View Go Run Terminal Help
Data.json - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
[ ] Person
{} 0
Name Shreeraj
Age 22
Mo.No: 8805334...
{} 1
Name Anand
Age 21
Mo.No: 8986854...
{} 2
Name Mayur
Age 20
Exp5.py
Data.json
FADP > Data.json > [ ] Person > {} 2 > Mo.No:
{
"Person":
[
{
"Name":"Shreeraj",
"Age":"22",
"Mo.No:":"8805334252"
},
{
"Name":"Anand",
"Age":"21",
"Mo.No:":"8986854123"
},
{
"Name":"Mayur",
"Age":"20",
"Mo.No:":"8963416895"
}
]
}
TERMINAL
2: Code
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp5.py"
{'Name': 'Shreeraj', 'Age': '22', 'Mo.No:': '8805334252'}
{'Name': 'Anand', 'Age': '21', 'Mo.No:': '8986854123'}
{'Name': 'Mayur', 'Age': '20', 'Mo.No:': '8963416895'}
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
raulshriraj@gmail.com
Live Share
Ln 19, Col 33
Spaces: 4
UTF-8
CRLF
JSON
Type here to search
ENG
02:43 PM
26-03-2021


File Edit Selection View Go Run Terminal Help
• Exp5.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
FADP
.vscode
launch.json
Data.json
Exp1.1.py
Exp1.py
Exp2.py
Exp3.py
Exp4.py
Exp5.py
Exp6.py
Exp7.py
Exp8.py
Exp9.py
Exp10.py
OUTLINE
json
f
data
i
Exp5.py
FADP > Exp5.py > ...
1 # Python program to read
2 # json file
3
4 import json
5 # Opening JSON file
6 f = open('Data.json')
7 # returns JSON object as
8 # a dictionary
9 data = json.load(f)
10 # Iterating through the json
11 # list
12 for i in data['Personcls']:
13 print(i)
14 # Closing file
15 f.close()
TERMINAL
2: Code
PS C:\Users\Shree\Desktop\FADP> python -u "c:\Users\Shree\Desktop\FADP\Exp5.py"
{'Name': 'Shreeraj', 'Age': '22', 'Mo.No:': '8805334252'}
{'Name': 'Anand', 'Age': '21', 'Mo.No:': '8986854123'}
{'Name': 'Mayur', 'Age': '20', 'Mo.No:': '8963416895'}
PS C:\Users\Shree\Desktop\FADP>
Python 3.9.2 64-bit
raulshriraj@gmail.com
Live Share
Ln 12, Col 25
Spaces: 4
UTF-8
CRLF
Python
Type here to search
ENG
02:43 PM
26-03-2021

Assignment 6

Problem Definition:

Write a python program to import charting library in python and perform following operations on it.

a) Creating a chart
b) Labeling the Axes

Theory:

This program will introduce you a graphing in python with matplotlib, which is arguably the most popular graphing and data visualization library for python.

Installation

Easiest way to install matplotlib is to use pip. Type following command in terminal:

pip install matplotlib

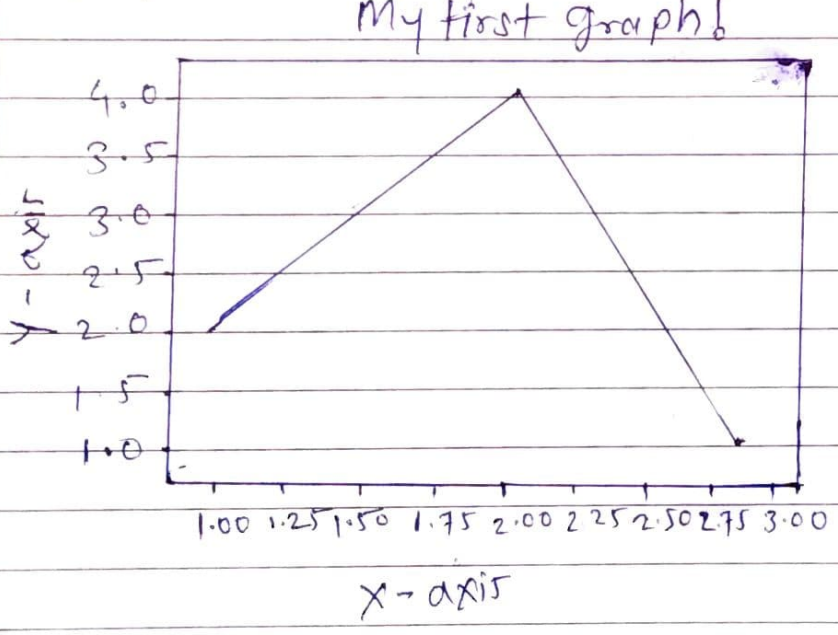
Getting started (plotting a line)

Code:

```
# importing the required module
import matplotlib.pyplot as plt

# x axis value
x = [1,2,3]
```

```
# corresponding Y axis values
Y = [2, 4, 1]

# plotting the points
plt.plot (x, y)

# naming the x-axis
plt.xlabel ('x-axis')

# naming the y axis
plt.ylabel ('y-axis')

# giving title to my graph
plt.title ('My first graph!')

# function to show the plot
plt.show()
```

Output:



The code seems self explanatory. following step were followed:

Define the x-axis and corresponding y-axis values as lists.
plot them on canvas using .plot() function.
Give a name to x-axis and y-axis using .Xlabel() and .Ylabel() functions. Give a title to your plot using .title() functions
Finally, to view your plot, we use .show() function

Conclusion:

Hence we have implemented python program to import charting library in python and perform following operations on it

a) Creating a chart.
b) Labeling the Axes.

File Edit Selection View Go Run Terminal Help — Exp6.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- launch.json
- Data.json
- Exp1.1.py
- Exp1.py
- Exp2.py
- Exp3.py
- Exp4.py
- Exp5.py
- Exp6.py
- Exp7.py
- Exp8.py
- Exp9.py
- Exp10.py
- students.xlsx
- tempCodeRunnerFil...

OUTLINE

- plt
- x
- y

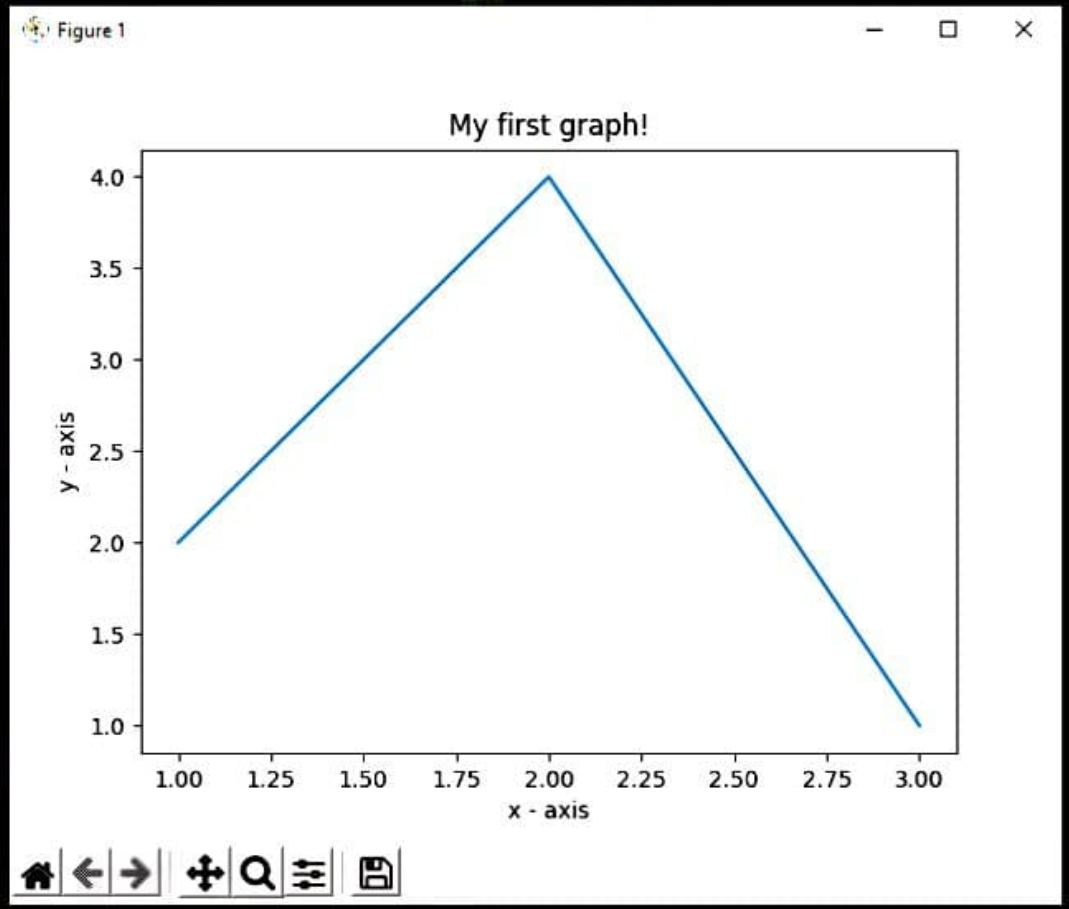
Exp6.py

FADP > Exp6.py > ...

```
1  # importing the required module
2  import matplotlib.pyplot as plt
3  # x axis values
4
5  x = [1,2,3]
6  # corresponding y axis values
7  y = [2,4,1]
8  # plotting the points
9  plt.plot(x, y)
10 # naming the x axis
11 plt.xlabel('x - axis')
12 # naming the y axis
13 plt.ylabel('y - axis')
14 # giving a title to my graph
15 plt.title('My first graph!')
16 # function to show the plot
17 plt.show()
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python
-u "c:\Users\Shree\Desktop\FADP\Exp6.py
"
```



Python 3.9.2 64-bit ⊗ 0 ⚠ 0 raulshriraj@gmail.com Live Share — Ln 8, Col 22 Spaces: 4 UTF-8 CRLF Python

Type here to search

ENG 02:44 PM 26-03-2021

Assignment: 7

Problem Definition:

Perform following operations assignment no. 7

a) Formatting line type and color
b) saving the chart file

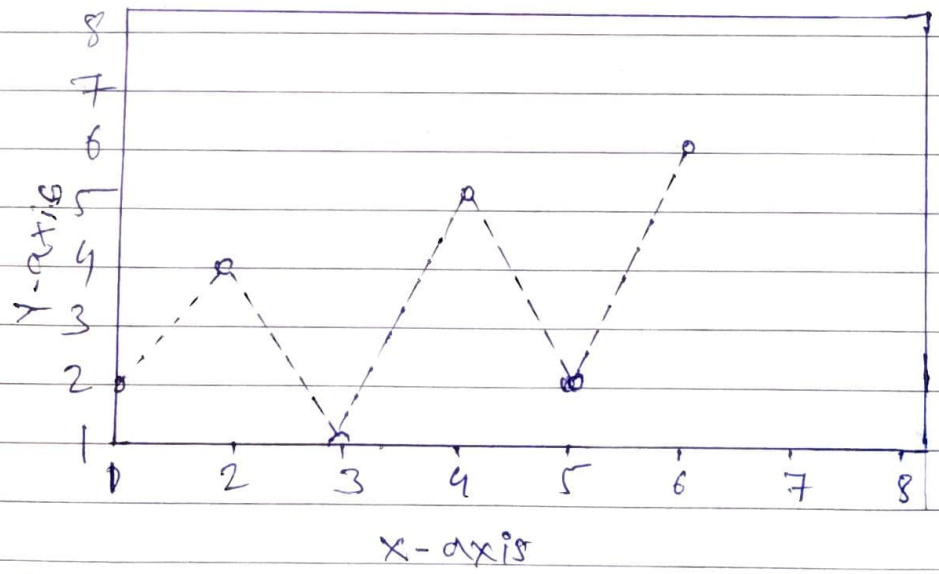
Theory:

Customization of plots
Here, we discuss some elementry customizations applicable on almost any plot

```
import matplotlib.pyplot as plt

# X axis values
X = [1,2,3,4,5,6]

# Corresponding Y axis values
y = [2,4,1,5,2,6]

# plating the points

plt.plot (X, Y, colour = "green', linestyle = 'dashed',
        linewidth = 3, marker = 'o', markerfacecolor =
        'blue', markersize = 12)

# Setting X and Y axis range
plt.ylim(1,8)
plt.xlim(1,8)
```

```
# naming the x axis
plt.xlabel('x-axis')

# naming the y axis
plt.ylabel('y-axis')

# giving a title to my graph
plt.title('some cool customizations!')

# function to show the plot
plt.show()
```

output :



As you can see, we have done several customi-zation like

Setting the line-width, line-style, line-color.

setting the marker, marker's face color, marker's size

overriding the X and Y axis range. If overriding is not done, pyplot module uses auto-scale feature to set the axis range and scale.

The savefig Method

With a simple chart under our belts, now we can opt to output the chart to a file instead of displaying it (or both if desired), by using the .savefig() method.

```
plt.savefig('books_read.png')
```

The .savefig() method requires a file name be specified as the first argument. This file name can be a full path and as seen above, can also include a particular file extension if desired. If no extension is provided, the configuration value of savefig.format is used instead.

Conclusion:

Hence we have implemented python program to perform following operations

a) Formatting Line type and colour
b) Saving the chart file

File Edit Selection View Go Run Terminal Help — Exp7.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- launch.json
- Data.json
- Exp1.1.py
- Exp1.py
- Exp2.py
- Exp3.py
- Exp4.py
- Exp5.py
- Exp6.py
- Exp7.py
- Exp8.py
- Exp9.py
- Exp10.py
- students.xlsx
- tempCodeRunnerFil...

OUTLINE

- plt
- x
- y

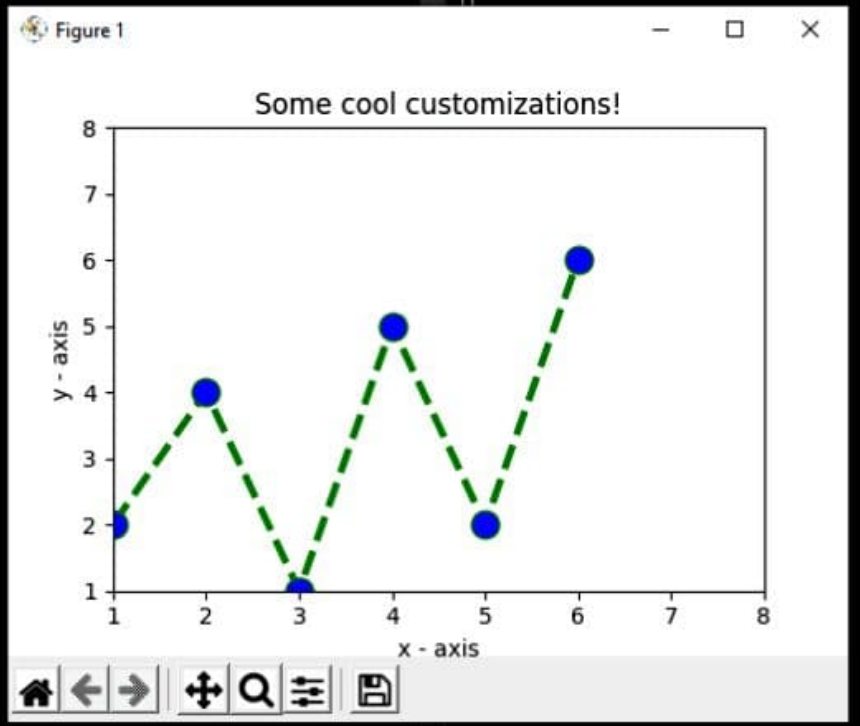
Exp7.py

FADP > Exp7.py > {} plt

```
1  import matplotlib.pyplot as plt
2  # x axis values
3  x = [1,2,3,4,5,6]
4  # corresponding y axis values
5  y = [2,4,1,5,2,6]
6  # plotting the points
7  plt.plot(x, y, color='green', linestyle='dashed', linewidth = 3,
8  marker='o', markerfacecolor='blue', markersize=12)
9
10 # setting x and y axis range
11 plt.ylim(1,8)
12 plt.xlim(1,8)
13 # naming the x axis
14 plt.xlabel('x - axis')
15 # naming the y axis
16 plt.ylabel('y - axis')
17 # giving a title to my graph
18 plt.title('Some cool customizations!')
19 # function to show the plot
20 plt.show()
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python
-u "c:\Users\Shree\Desktop\FADP\Exp7.py
"
```



Python 3.9.2 64-bit ⊗ 0 ⚠ 0 raulshriraj@gmail.com Live Share — Ln 1, Col 32 Spaces: 4 UTF-8 CRLF Python

Type here to search

ENG 02:44 PM 26-03-2021

Date
Page 21

Assignment 8

Problem definition:
Perform Following operations assignment no 8

a) Adding Annotations
b) Adding Legends
c) chart presentation style

Theory:

Matplotlib is one of the most popular python packages used for data visualization. It is a cross platform library for making 2D plot from data in arrays. pyplot is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g. creates a figure, creates a plotting area in figure, plot some lines in a plotting area, decorates the plot with labels etc

matplotlib.pyplot.legend()
A legend is an area describing the elements of the graph. In the matplotlib library, there's a function called legend() which is used to Place a legend on the axes.

The attribute Loc in legend() is used to specify the location of the legend. Default Value of loc is loc="best" (upper left). The string 'upper left', 'upper right', 'lower left', 'lower right' place the legend at the corresponding corner of the axes /figure.

The attribute bbox_to_anchor=(X, Y) of legend() function is used to specify the coordinates of the legend, and the attribute ncol represents the number of columns that the legend has. Its default value is 1

Syntax:

matplotlib.pyplot.legend(["blue", "green"], bbox_to_anchor=(0.75, 1.15), ncol=2)

The following are some more attributes of function legend():

shadow: [None or bool] Whether to draw a shadow behind the legend. Its Default Value is none.

Markerscale: [None or int or float] The relative size of legend markers compared with the originally drawn ones. The Default is None.

numpoints: [None or int] The number of marker point in the legend when creating a legend entry for a Line2D (Line). The Default is None.

fontsize: The font size of legend. If the value is numeric the size will be the absolute font size in points.

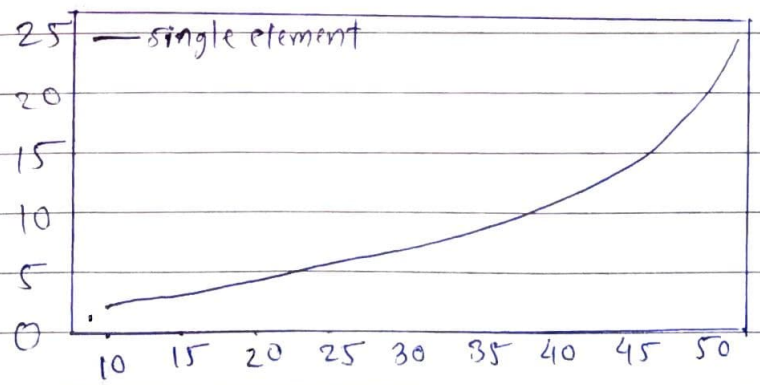
facecolor: [None or "inherit" or color] The legend's background color.

edgecolor : [None or "inherit" or color] The legend's back-ground patch edge color.

Ways to use legend() function in python -

Example:

```
import numpy as np
import matplotlib.pyplot as plt

# Xaxis values
X = [1,2,3,4,5]

# Y-axis values
Y = [1,4,9,16,25]

# Function add to plot
plt.plot(x,Y)

# Function add a legend
plt.legend(['single element'])

# function to show the plot
plt.show()
```



Conclusion:

Hence we have implemented python program to perfor-m following operations.

a) Adding Annotations
b) Adding Legends
c) Chart presentation style.

File Edit Selection View Go Run Terminal Help — Exp8.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- launch.json
- Data.json
- Exp1.1.py
- Exp1.py
- Exp2.py
- Exp3.py
- Exp4.py
- Exp5.py
- Exp6.py
- Exp7.py
- Exp8.py
- Exp9.py
- Exp10.py
- students.xlsx
- tempCodeRunnerFil...

OUTLINE

- np
- plt
- x
- y

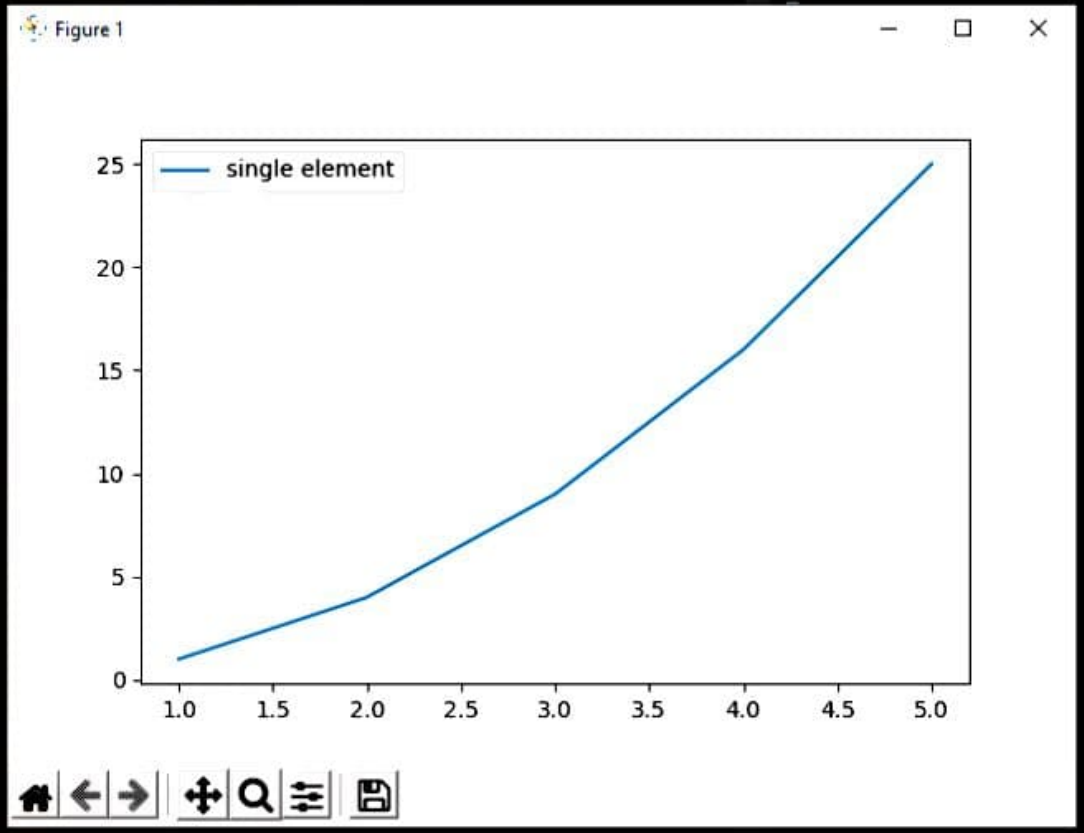
Exp8.py

FADP > Exp8.py > ...

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  # X-axis values
4  x = [1, 2, 3, 4, 5]
5  # Y-axis values
6  y = [1, 4, 9, 16, 25]
7  # Function to plot
8  plt.plot(x, y)
9  # Function add a legend
10 plt.legend(['single element'])
11 # function to show the plot
12 plt.show()
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python
-u "c:\Users\Shree\Desktop\FADP\Exp8.py
"
```



Python 3.9.2 64-bit ⊗ 0 ⚠ 0 raulshriraj@gmail.com Live Share — Ln 12, Col 11 Spaces: 4 UTF-8 CRLF Python

Type here to search

ENG 02:45 PM 26-03-2021

Assignment 9

Date
Page 25

Problem Definition:
write a python program using data Analytics libraries to plot a bar chart for given data.

Theory:

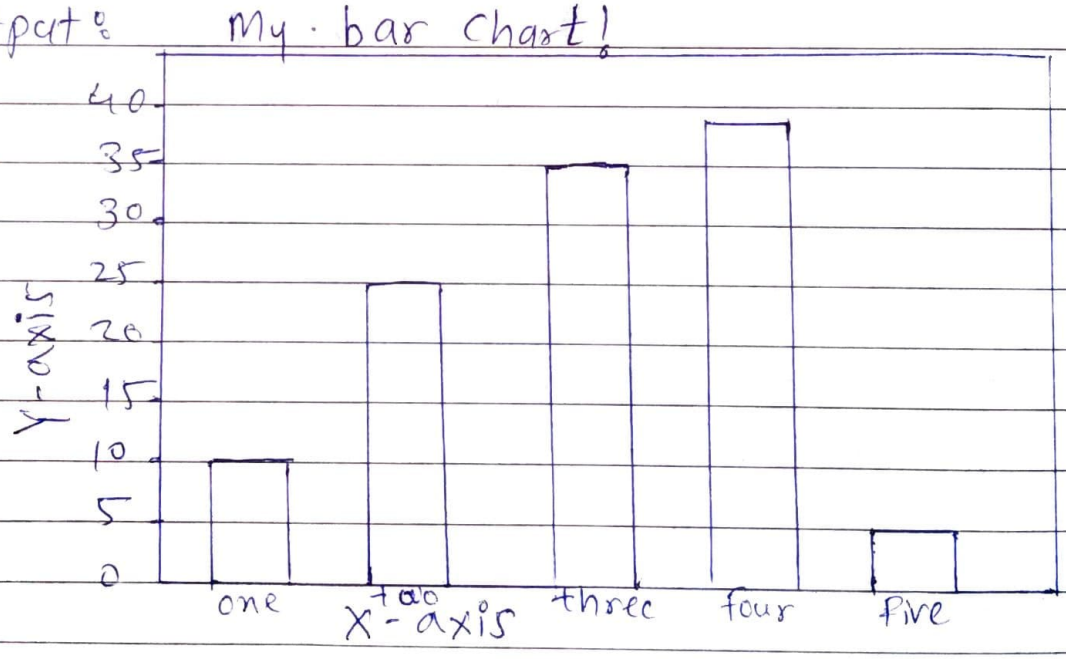
Bar chart:
A bar chart to bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.

A bar graph show comparisons among discrete categories. One axis of the chart shows the specific categories being compared, and the other axis represents a measured value. Some bar graphs present bars clustered in group of more than one, showing the value of more than one measured variable.

Code:
```
import matplotlib.pyplot as plt

#X-coordinates of left side of bars
left = [1,2,3,4,5]

#height of bars
height = [10,24,36,40,45]

# labels for bars
tick_label = ['one','two','three','Four','Five']
```

```
# plotting a bar chart
plt.bar(left, height, tick_label = tick_label, width =
    0.8, color = ['red', 'green'])

# naming the x-axis
plt.xlabel('x-axis')

# naming the y-axis
plt.ylabel('y_axis')

# plot.title
plt.title('My bar chart!')

# function to show the plot
plt.show
```

Output:



Conclusion: Hence we have implemented python program using data analytic libraries to plot a bar chart for given data.

File Edit Selection View Go Run Terminal Help — Exp9.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- Data.json
- Exp1.1.py
- Exp1.py
- Exp2.py
- Exp3.py
- Exp4.py
- Exp5.py
- Exp6.py
- Exp7.py
- Exp8.py
- Exp9.py
- Exp10.py
- students.xlsx
- tempCodeRunnerFil...

OUTLINE

- plt
- left
- height
- tick_label

Exp9.py

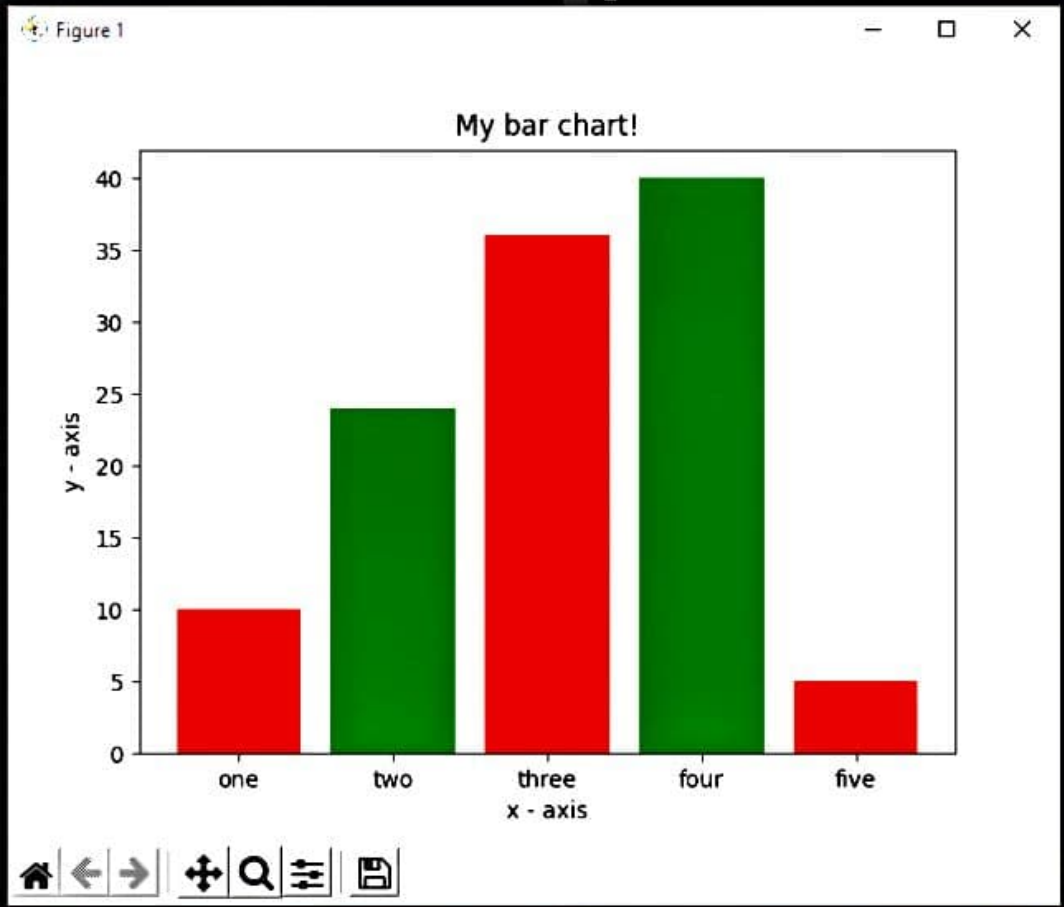
FADP > Exp9.py > ...

```
1  import matplotlib.pyplot as plt
2  # x-coordinates of left sides of bars
3  left = [1, 2, 3, 4, 5]
4  # heights of bars
5  height = [10, 24, 36, 40, 5]
6  # labels for bars
7  tick_label = ['one', 'two', 'three', 'four', 'five']
8  # plotting a bar chart
9  plt.bar(left, height, tick_label = tick_label,
10 width = 0.8, color = ['red', 'green'])
11 # naming the x-axis
12 plt.xlabel('x - axis')
13 # naming the y-axis
14 plt.ylabel('y - axis')
15 # plot title
16 plt.title('My bar chart!')
17 # function to show the plot
18 plt.show()
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python
-u "c:\Users\Shree\Desktop\FADP\Exp9.py
"
```

Figure 1



Python 3.9.2 64-bit ⊗ 0 ⚠ 0 raulshriraj@gmail.com Live Share — Ln 18, Col 11 Spaces: 4 UTF-8 CRLF Python

Type here to search

ENG 02:46 PM 26-03-2021

Assignment 10

Date
Page 27

Problem Definition:
Write a python program using data analytics libraries to plot heat map for given data

Theory:
A 2-D Heatmap is a data visualization tool that helps to represent the magnitude of the phenomeno-n in form of colors. In python we can plot 2-D Heatma-ps using Matplotlib package. There are different methods to plot 2-D Heatmaps, some of them are discussed below.

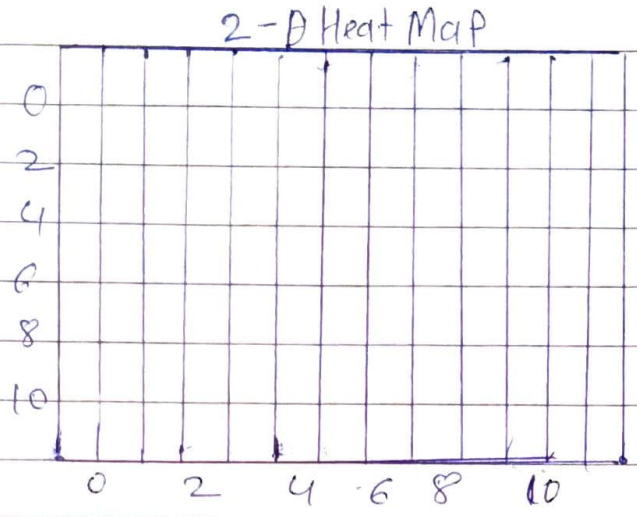
Method using matplotlib.pyplot.imshow() function

Syntax: matplotlib.pyplot.imshow(X, Cmap=None, norm=None, interpolation=None, alpha=None, Vmin=None, Vmax=None, origin=None, extent=None, shape=<deprecated parameter>, Filternorm=1, Filterrad=4.0, imlim=<deprecated parameter>, resamle=None, Url=None, *, data=None * * kwargs)

Code:

```
# Program to plot 2D Heat map
# using matplotlib.pyplot.imshow() method
import numpy as np
import matplotlib.pyplot as plt

data = np.random.random((12, 12))
plt.imshow(data, cmap='autumn', interpolation='nearest')

plt.title("2-D Heat Map")
plt.show()
```

Output:



Conclusion:

Hence we have implemented python program using data analytics libraries to plot a heat map for given data.

File Edit Selection View Go Run Terminal Help — Exp10.py - Untitled (Workspace) - Visual Studio Code

EXPLORER

UNTITLED (WORKSPACE)

- Data.json
- Exp1.1.py
- Exp1.py
- Exp2.py
- Exp3.py
- Exp4.py
- Exp5.py
- Exp6.py
- Exp7.py
- Exp8.py
- Exp9.py
- Exp10.py
- students.xlsx
- tempCodeRunnerFil...

OUTLINE

- {} np
- {} plt
- data

Exp10.py

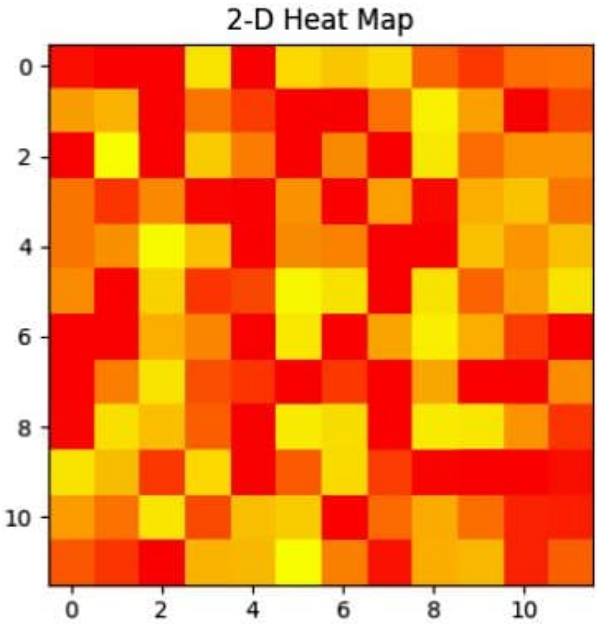
FADP > Exp10.py > ...

```
1  # Program to plot 2-D Heat map
2  # using matplotlib.pyplot.imshow() method
3  import numpy as np
4  import matplotlib.pyplot as plt
5  data = np.random.random(( 12 , 12 ))
6  plt.imshow( data , cmap = 'autumn' , interpolation = 'nearest' )
7  plt.title( "2-D Heat Map" )
8  plt.show()
```

TERMINAL — 2: Code

```
PS C:\Users\Shree\Desktop\FADP> python
-u "c:\Users\Shree\Desktop\FADP\Exp10.p
y"
```

Figure 1



Python 3.9.2 64-bit ⊗ 0 ⚠ 0 raulshriraj@gmail.com Live Share — Python

Type here to search

ENG 02:47 PM 26-03-2021