**Experiment – 11: Python Packages**

**1. Aim**: To implement a Python program to demonstrate creation and usage of Python packages

**2. Objectives:** After performing this experiment, the student will be able to create packages in Python and use them how to use them in programming.

**3. Outcomes:** Students shall be able to understand and apply Python packages and file handling. (LO 404.4)

**4. Prerequisite**: Knowledge of Python basic programming

**5. Requirements**: Personal Computer (PC), Windows /Linux Operating System, Python IDE

**6. Pre-Experiment Exercise:**

**Theory:**

Functions, modules and packages are all constructs in Python that promote code modularization. A module can contain multiple objects, such as classes, functions etc. A package can contain one or more relevant modules. Physically, a package is actually a folder containing one or more module files. Python has several in-built packages like Numpy, OpenCV, Pandas etc. However, developers can create their own packages and use/distribute them easily. Creating and accessing packages in Python requires 3 major steps.

i. Create a directory and include a \_\_init\_\_.py file in it to tell Python that the current directory is a package.

ii. Include other sub-packages or files you want.

iii. Access the sub-packages with the valid import statements.

**7. Laboratory Exercise**

**A. Procedure**

i. Open Python IDE for Python programming.

ii. Install numpy package in Python.

ii. Open new Python file from menu file-new.

iii. Type python code with proper syntax.

iv. Save file with .py extension.

v. Execute the file.

**B. Program code with comments:**

1.Create a Python package “myPackage”. Create two modules – mod1 and mod2 inside myPackage. Let mod1 contain a function to find addition of two numbers while mod2 contains a function to subtract two numbers. Write a Python program to import myPackage and test it.

**Code:**

Let's create a package named package1, using the following steps:

1.Create a new folder named D:Applications\Python\MyApp.

2.Inside ‘MyApp’, create a sub-directory named ‘package1’.

3.Create an empty \_\_init\_\_.py file in the ‘package1’ folder.

4.Using Python IDLE, create two modules ‘add3.py’ and ‘sub3.py’ with the following respective codes.

add3.py

def additn(a,b,c):

return (a+b+c)

sub3.py

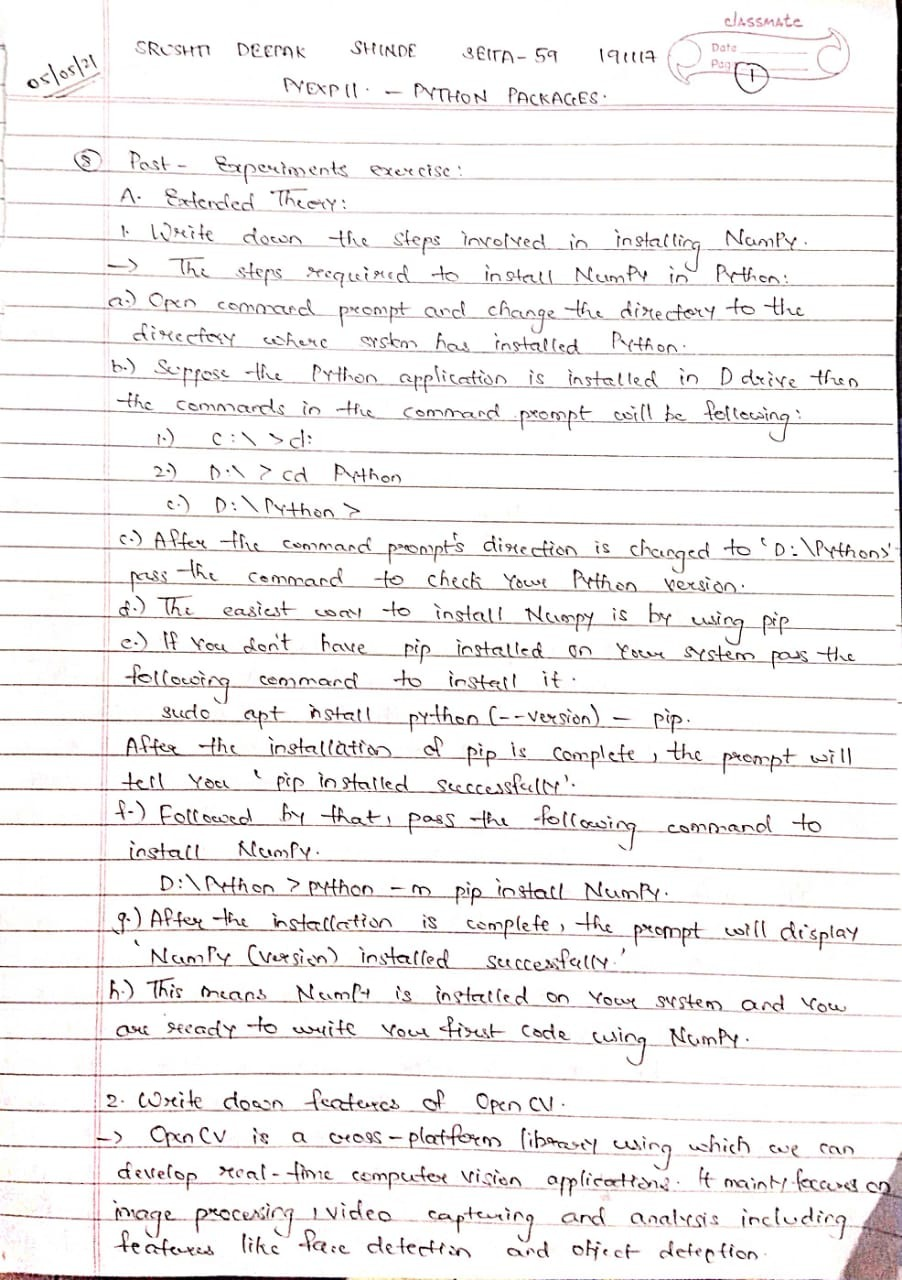
def subtn(a,b,c):

return (a-b-c)

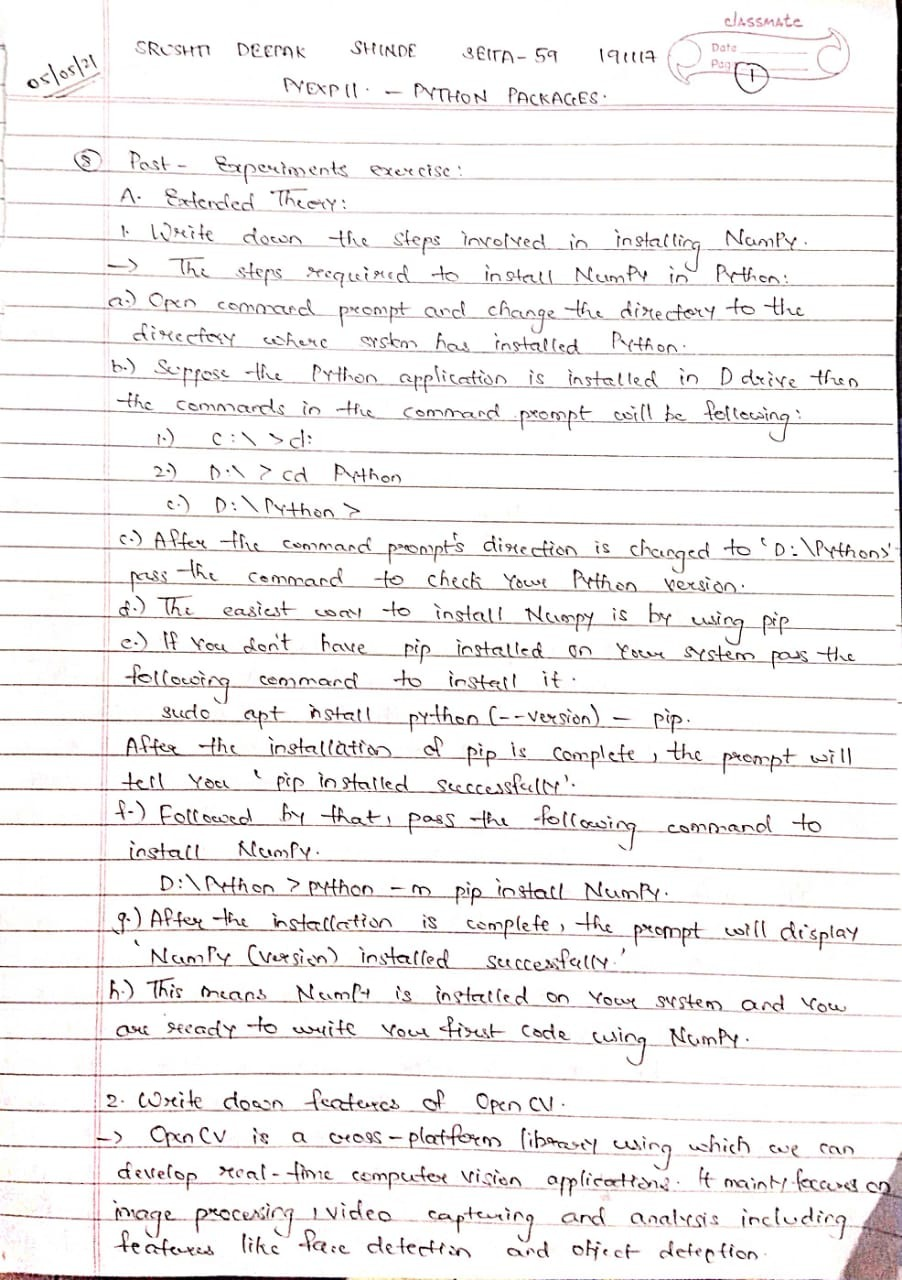
**8. Post-Experiments Exercise**

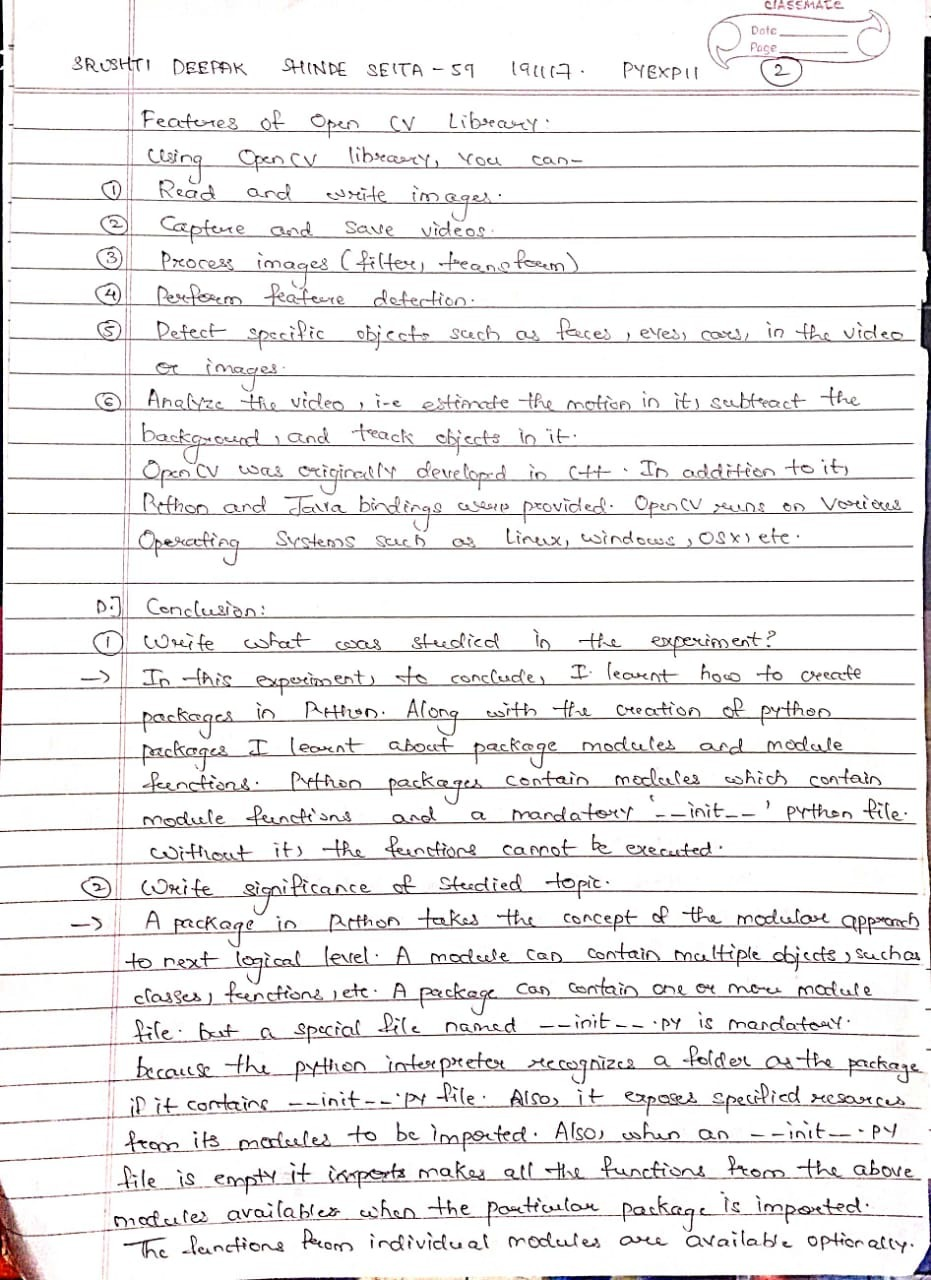
**A. Extended Theory:**

1. Write down the steps involved in installing NumPy.



2. Write down features of OpenCV.





**B. Results/Observations/Program output:**

Present the program input/output results and comment on the same.

**Output:**

Follow the following steps to run the modules created in python packages:

1.Open CMD and type ‘d:’, this will take the command prompt to the D drive where the python ‘MyApp’ folder is situated.

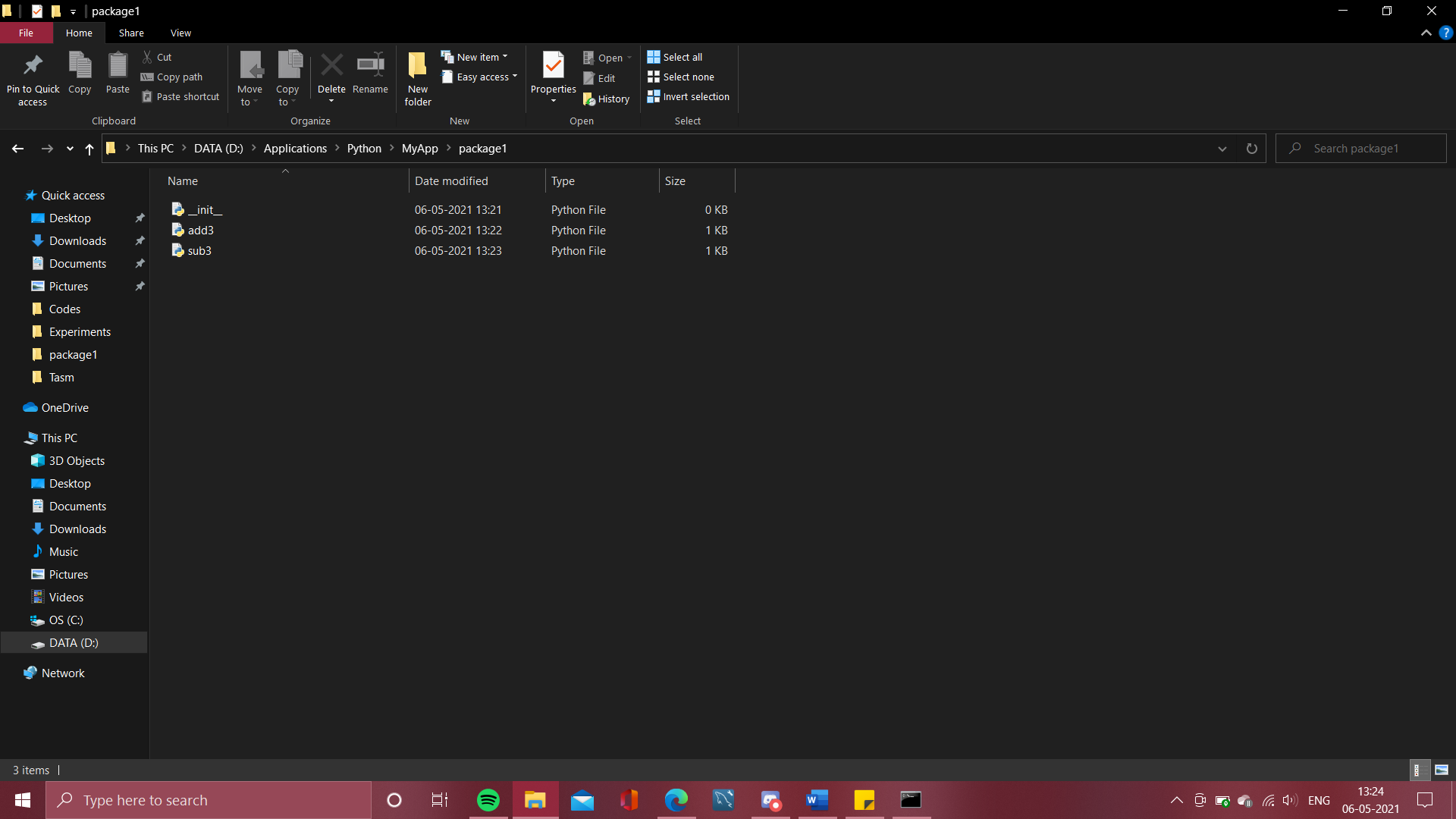
2.Change direction of CMD by

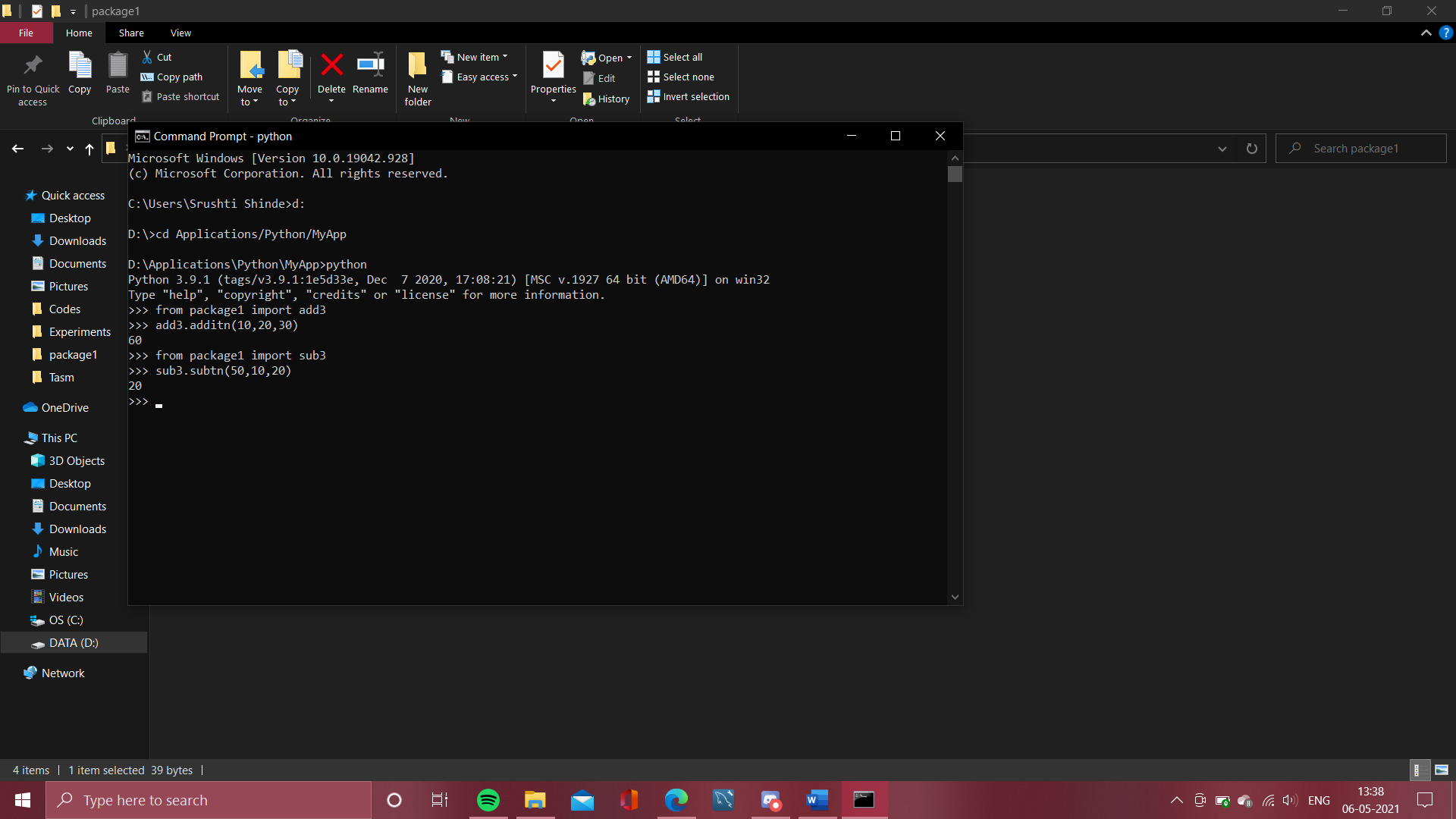
‘D:\>cd (copy location of MyApp folder)’

3.Invoke the python prompt

4.From the package1 import the modules.

5.Call the modules functions and execute.





**C. Questions/Programs:**

1. Write a Python program to find LCM and GCD of three numbers using Numpy .

**Code:**

import numpy as np

#for GCD

#create empty list

list1 = []

print('-----GCD of 3 Nummbers-----')

print('-->Enter the numbers-')

for item in range(0,3):

a=int(input())

#append to list1

list1.append(a)

#to find the highest common factor from the three items of the list1

gcd1=np.gcd.reduce(list1)

print(f'GCD = {gcd1}')

#for LCM

#create another empty list

list2 = []

print('-----LCM of 3 Numbers-----')

print('-->Enter the numbers:-')

for item1 in range(0,3):

b=int(input())

#append to list2

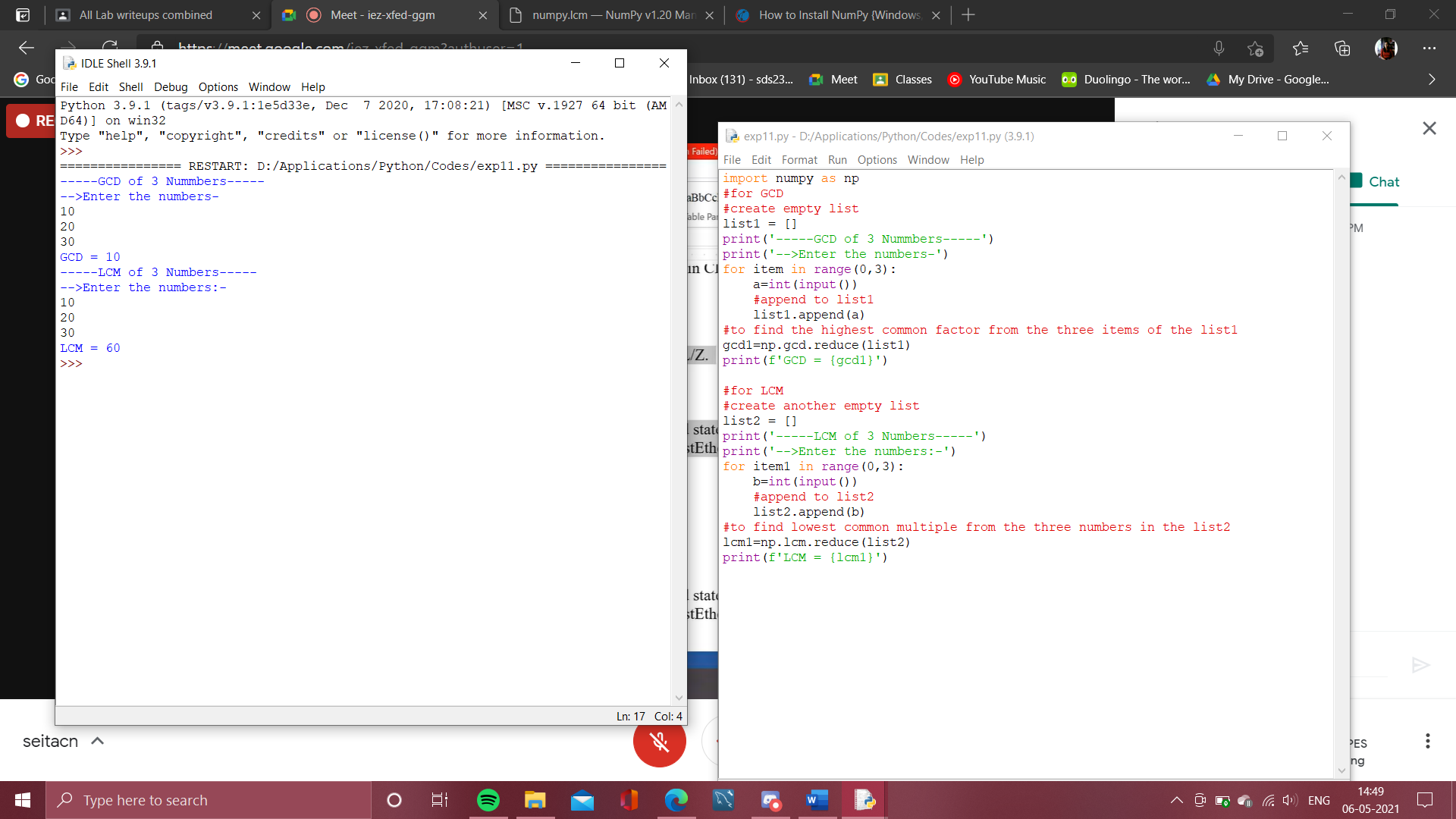
list2.append(b)

#to find lowest common multiple from the three numbers in the list2

lcm1=np.lcm.reduce(list2)

print(f'LCM = {lcm1}')

**Output:**

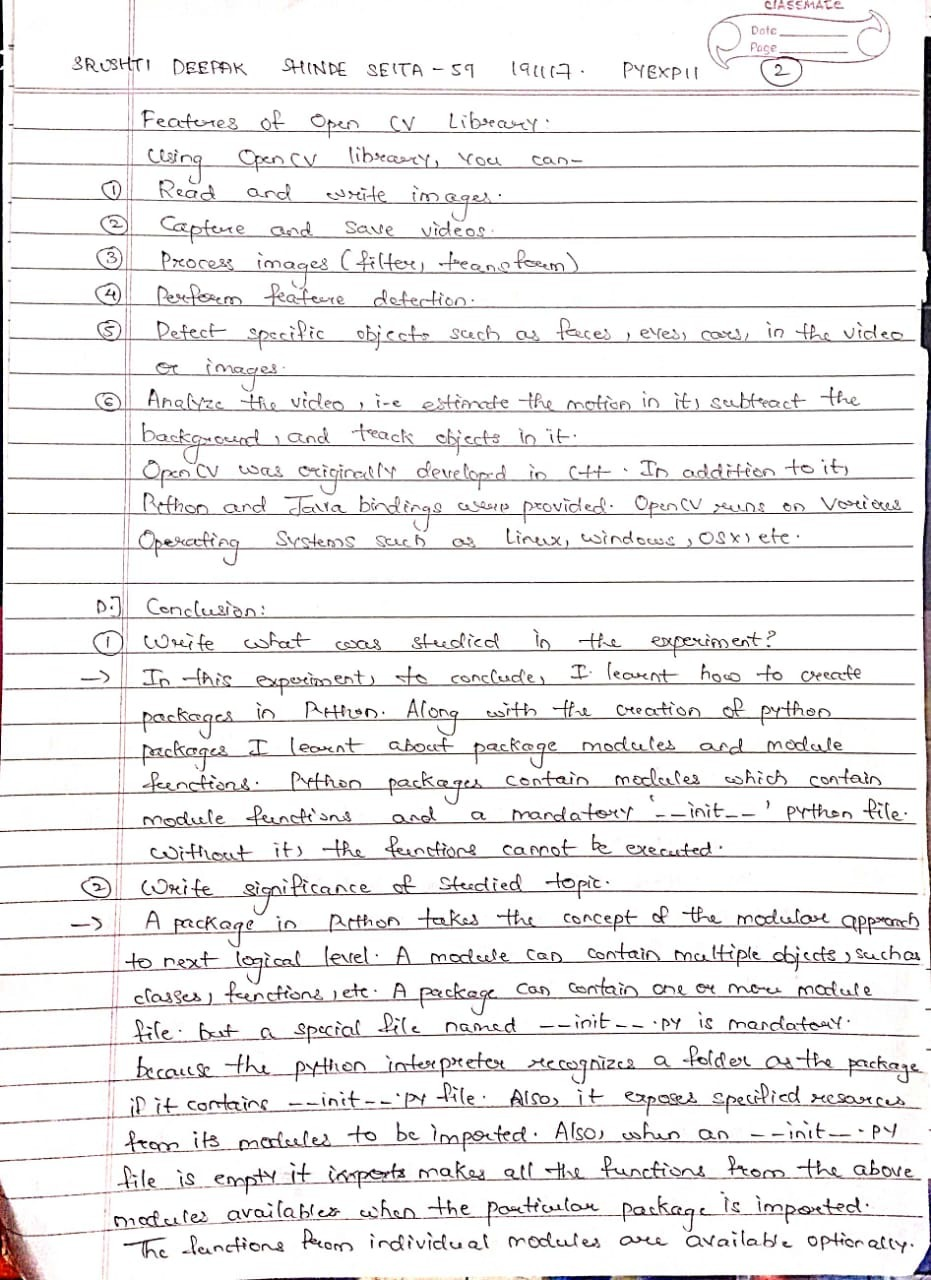


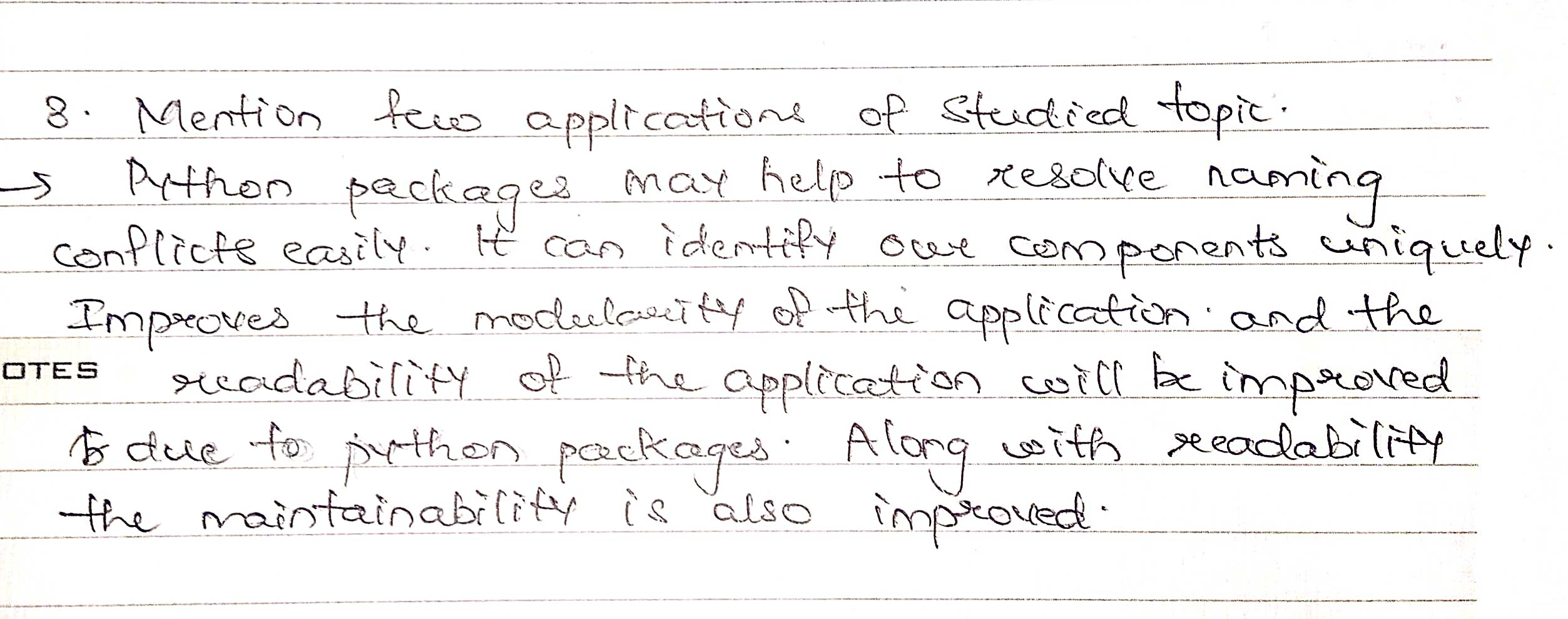
**D. Conclusion:**

1. Write what was performed in the experiment/program.

2. What is the significance of experiment/program?

3. Mention a few applications of what was studied.





**E. References**

[1] Mark Lutz, “Learning Python”, 5 th Edition, O’Reilly Publication

[2] <https://reconshell.com/best-python-libraries-and-packages-for-beginners/>

[3] <https://www.w3schools.com/python/numpy/numpy_ufunc_lcm.asp>