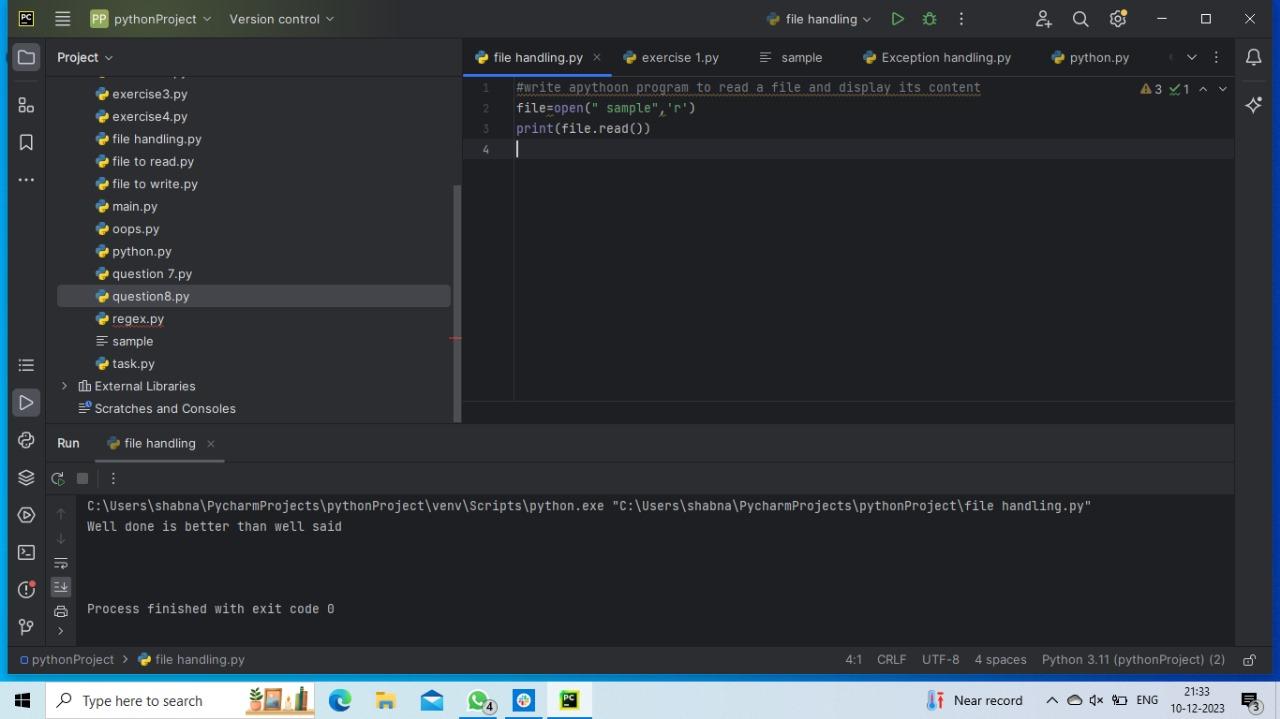
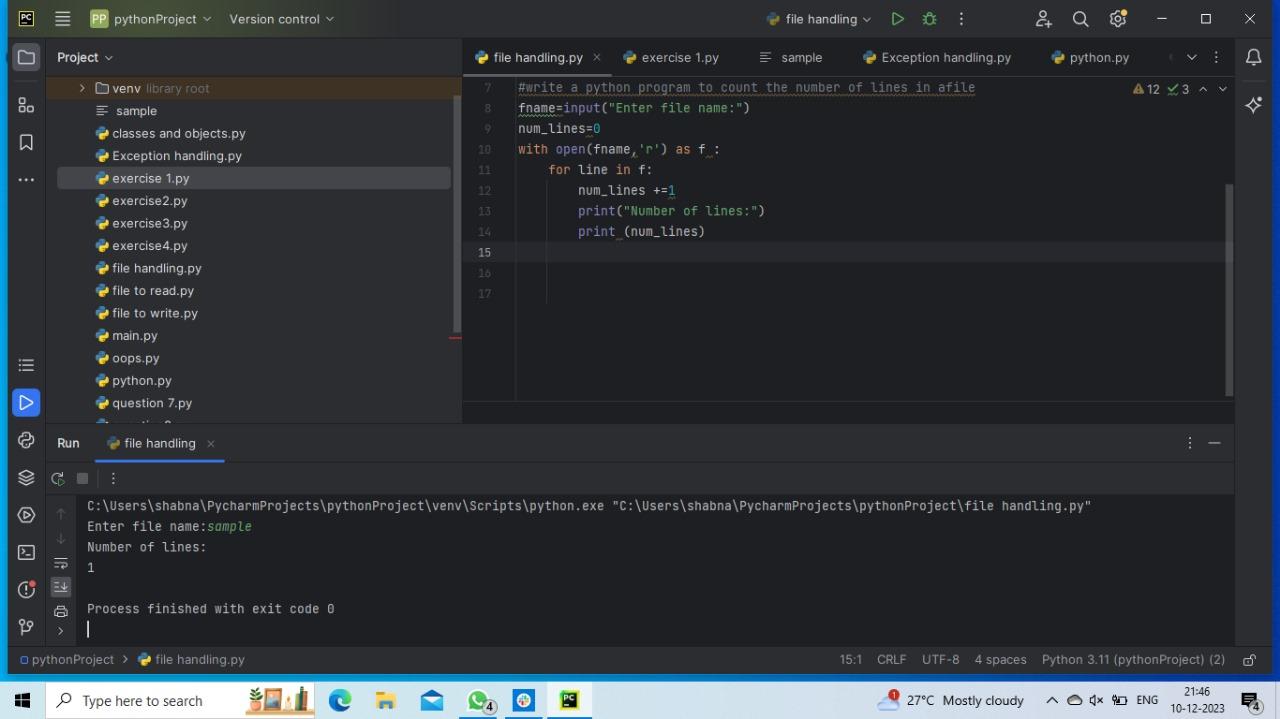
Task-9 [23-10-2023]

File Handling:

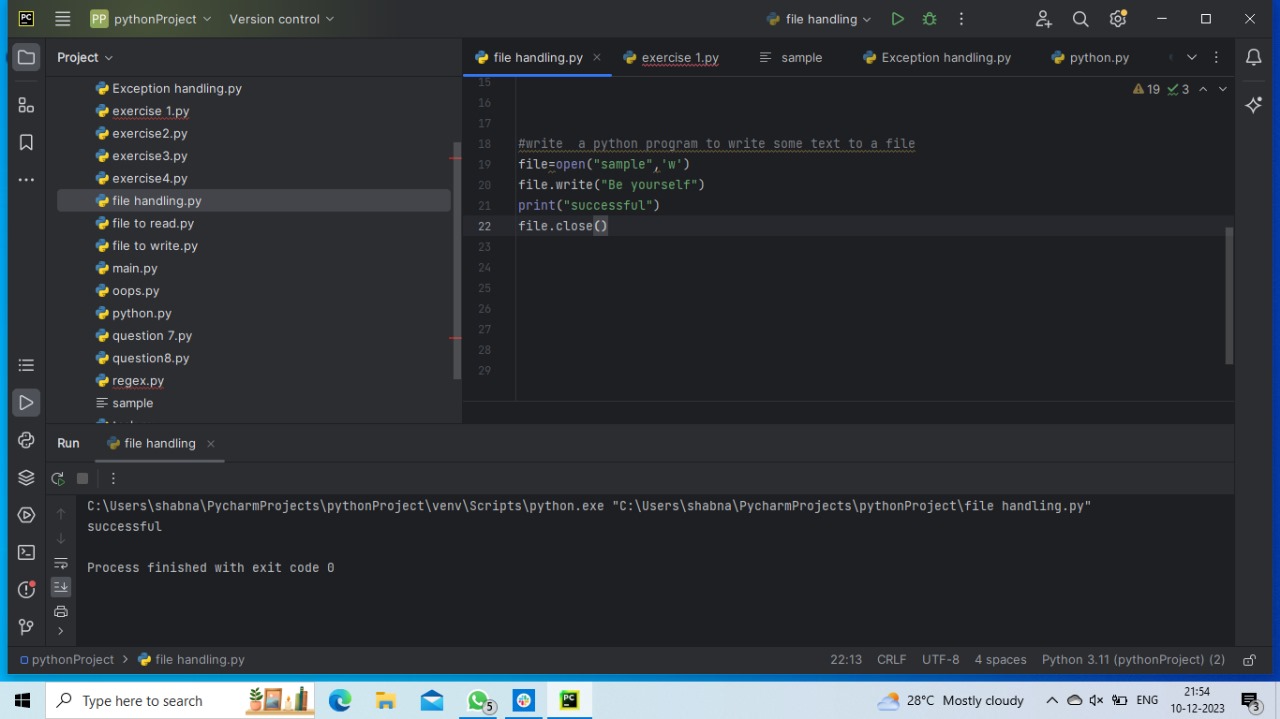
1.Write a Python program to read a file and display its contents



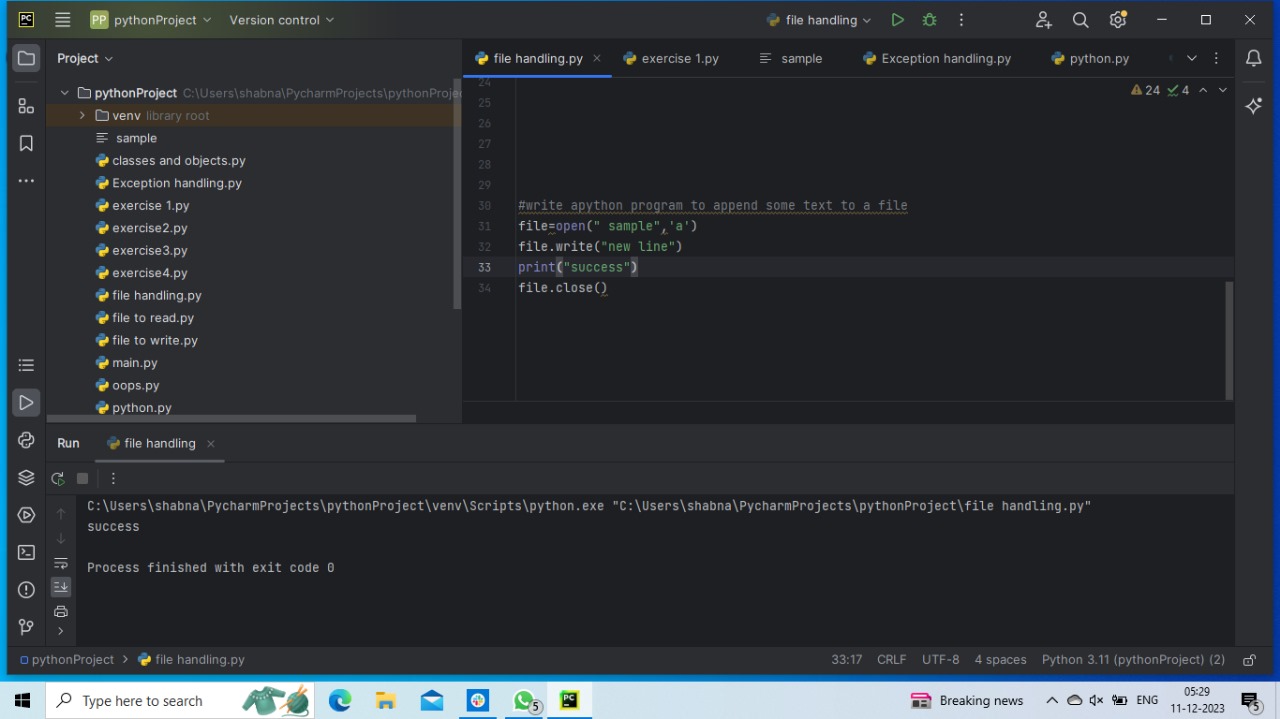
2.Write a Python program to count the number of lines in a file



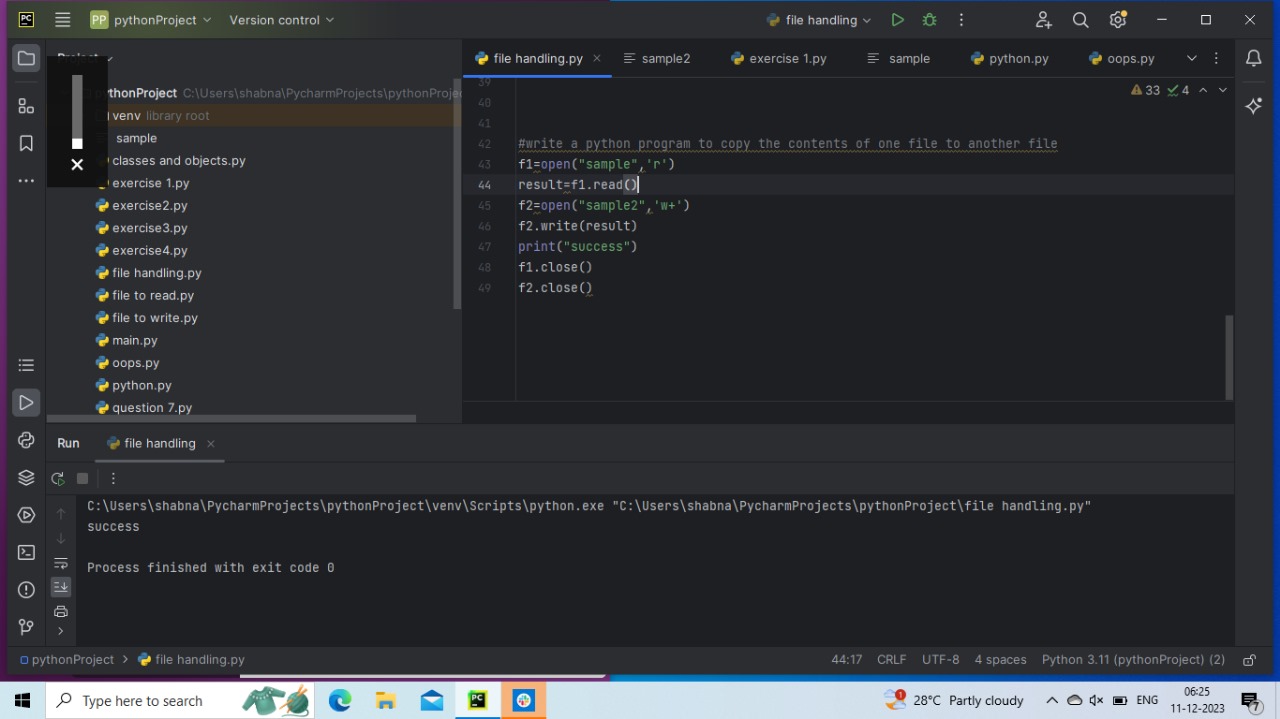
3.Write a Python program to write some text to a file



4.Write a Python program to append some text to a file.

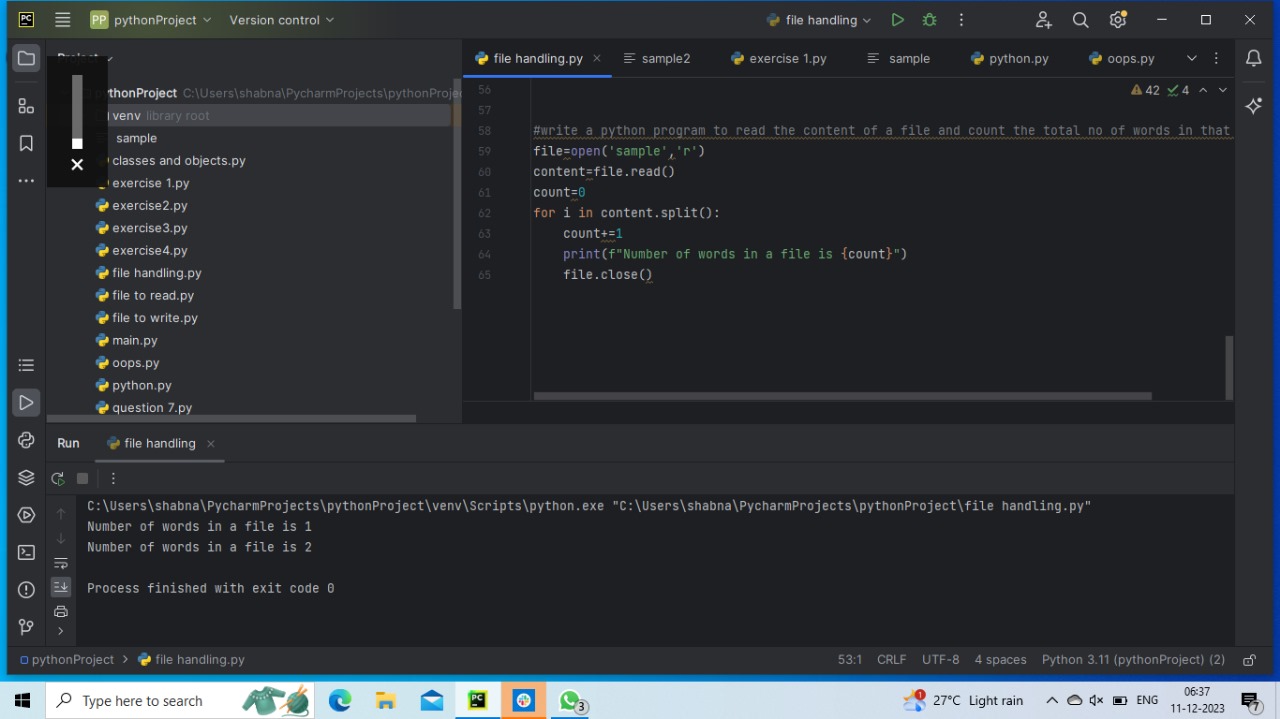


5.Write a Python program to copy the contents of one file to another file



6.Write a Python program to read the content of a file and count the total number

of words in that file.

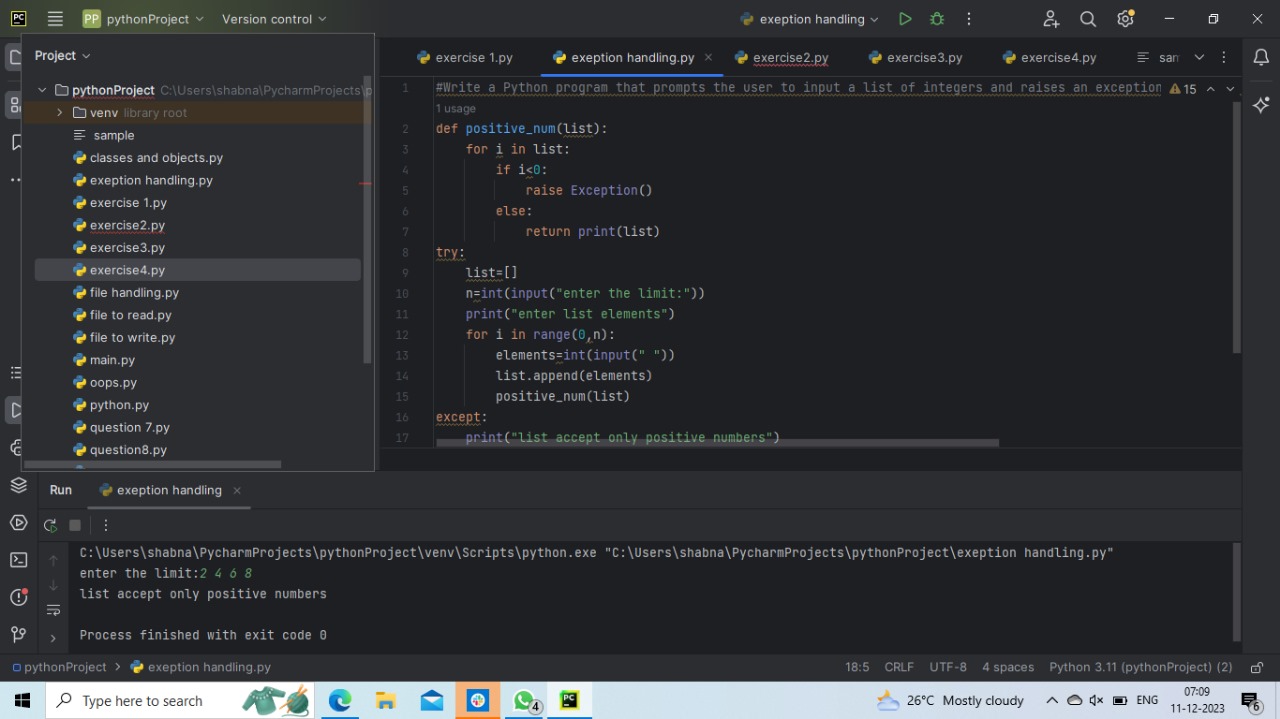


Task-10 [27-10-2023]

Exception Handling

1.Write a Python program that prompts the user to input a list of integers and

raises an exception if any of the integers in the list are negative.

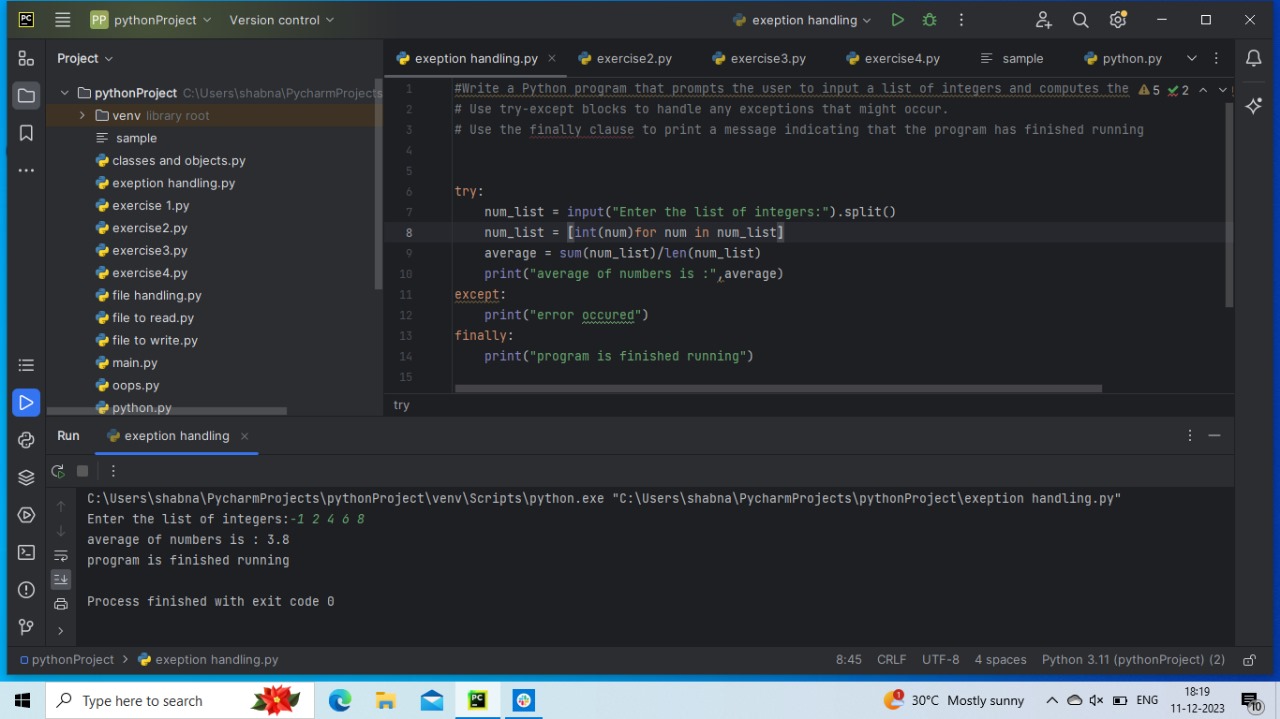


2.Write a Python program that prompts the user to input a list of integers and

computes the average of those integers. Use try-except blocks to handle any

exceptions that might occur. Use the finally clause to print a message indicating

that the program has finished running.



Task-11 [30-10-2023]

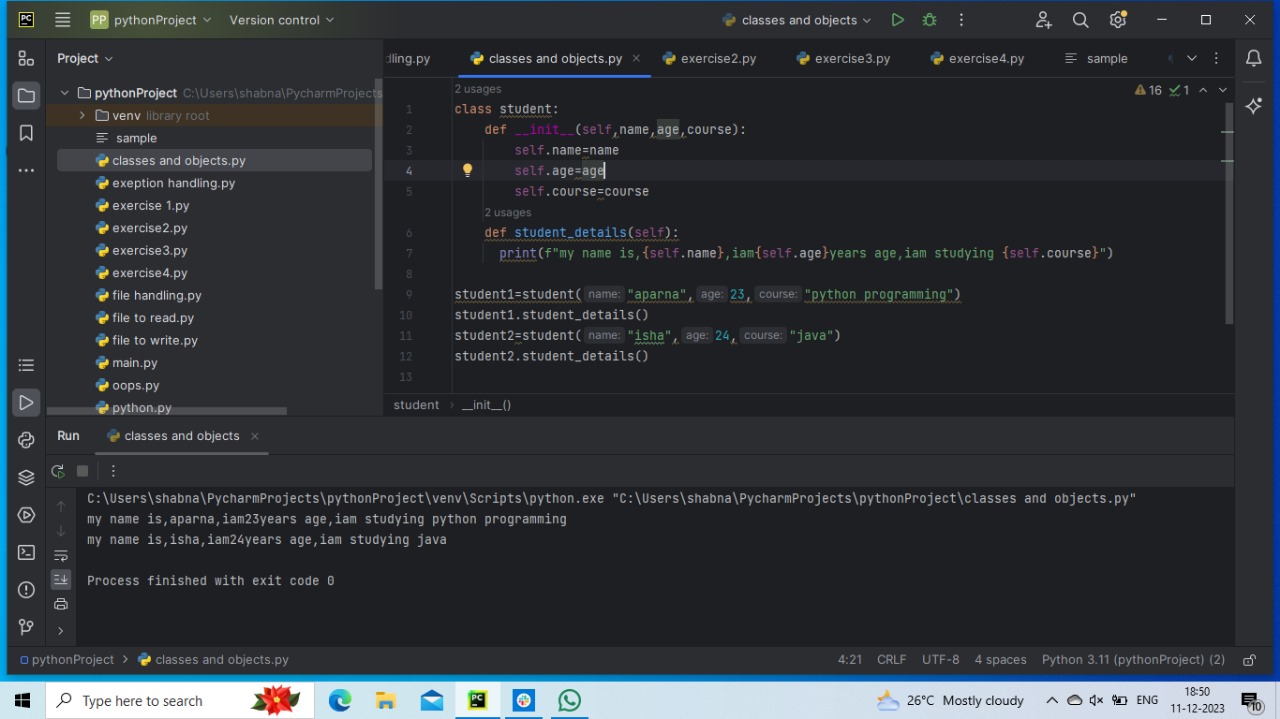
Classes and Objects:

1. Create a Python class called Student with a constructor that initializes attributes

for a student's name, age, and course. Then, write a method that allows a student to

introduce themselves. Finally, create two instances of the Student class and

demonstrate how to use them.



Task-12 [03-11-2023]

OOPs:

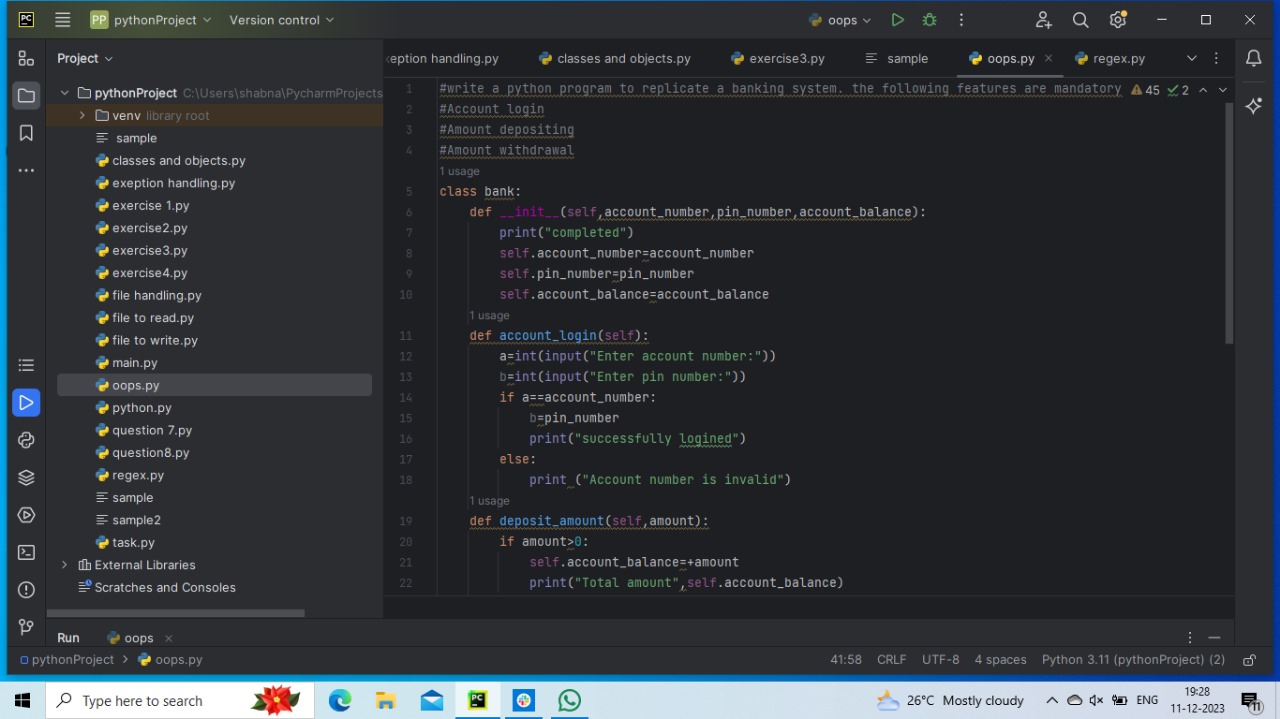
Write a python program to replicate a Banking system. The following features are

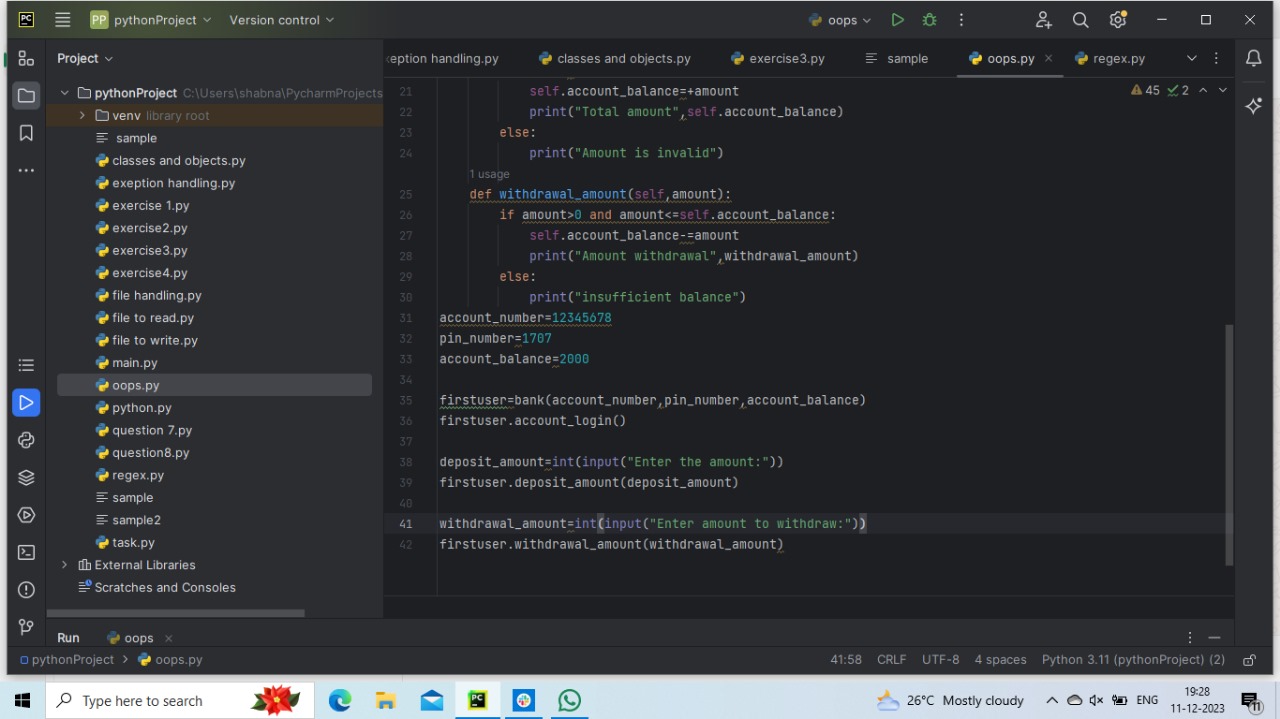
mandatory:

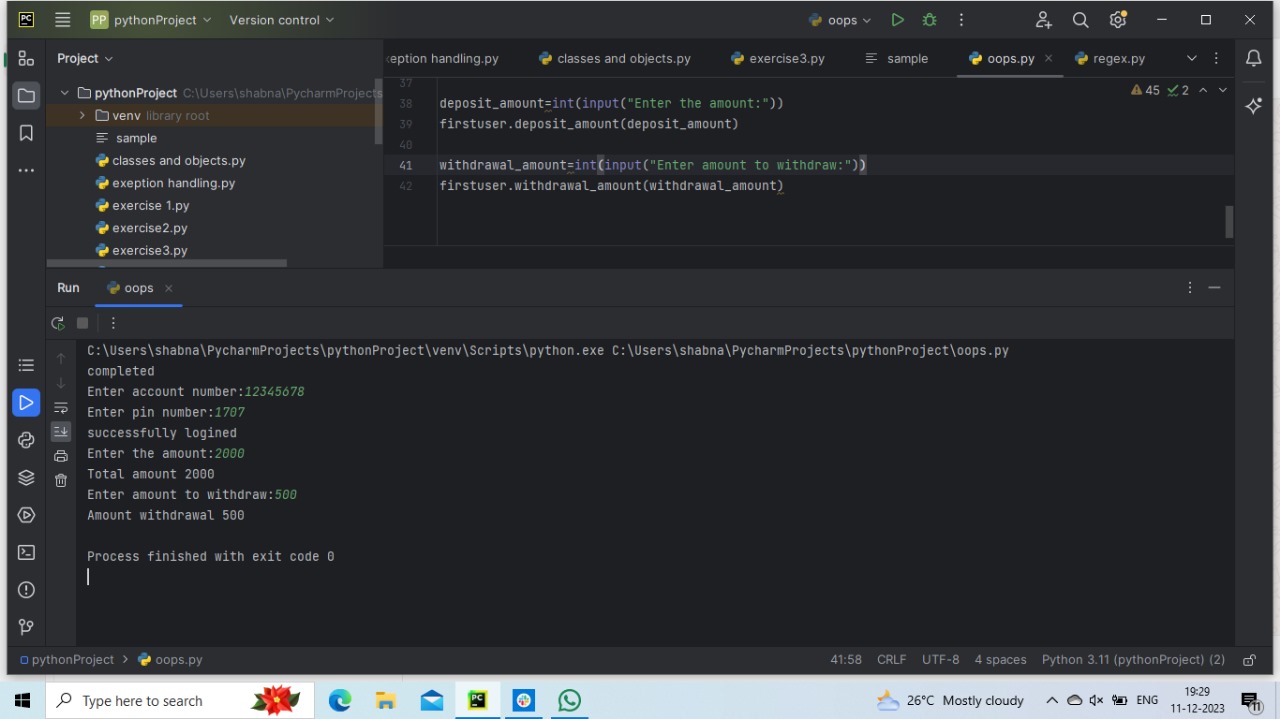
1.Account login

2. Amount Depositing

3. Amount Withdrawal



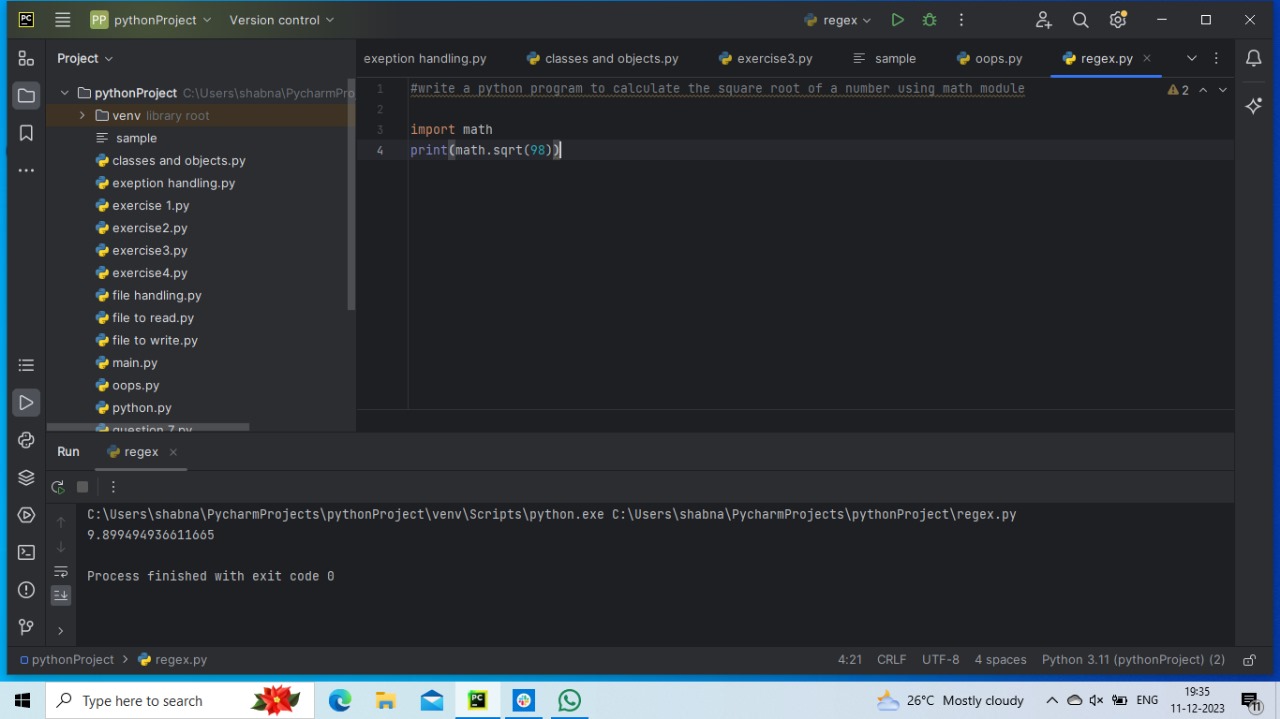




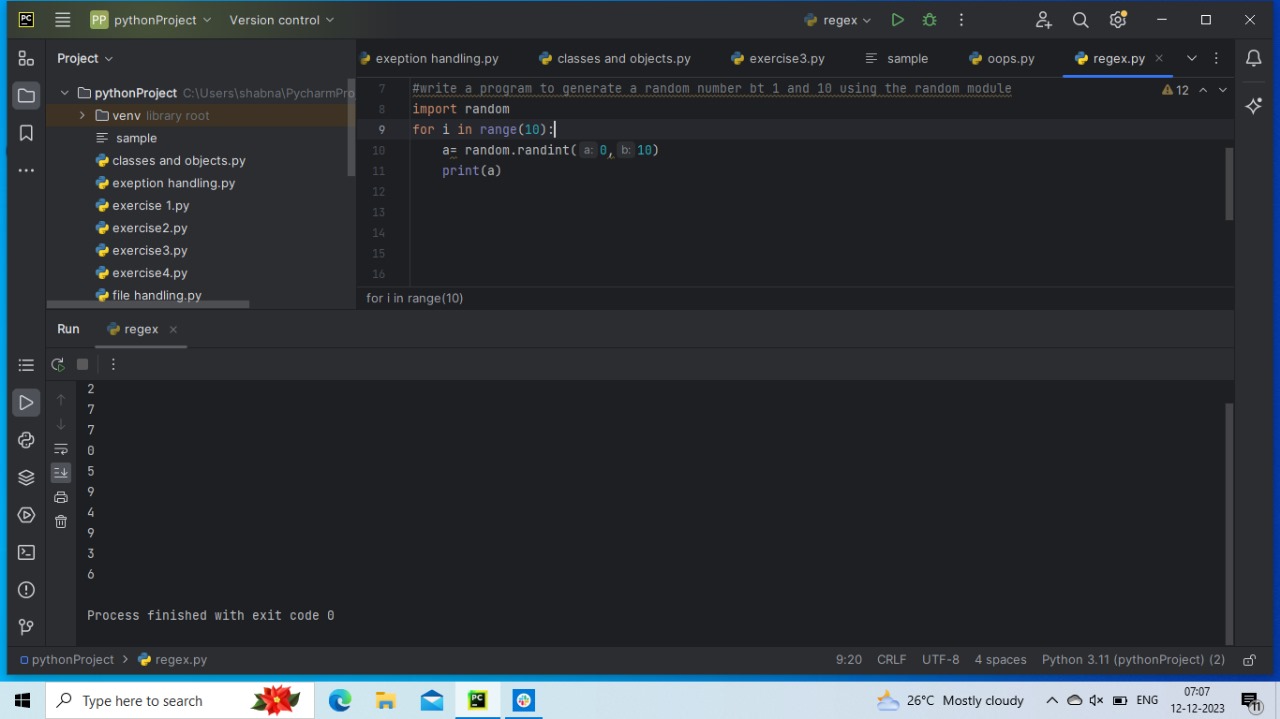
Task-13 [06-11-2023]

1.Write a Python program to calculate the square root of a number using the math

module.

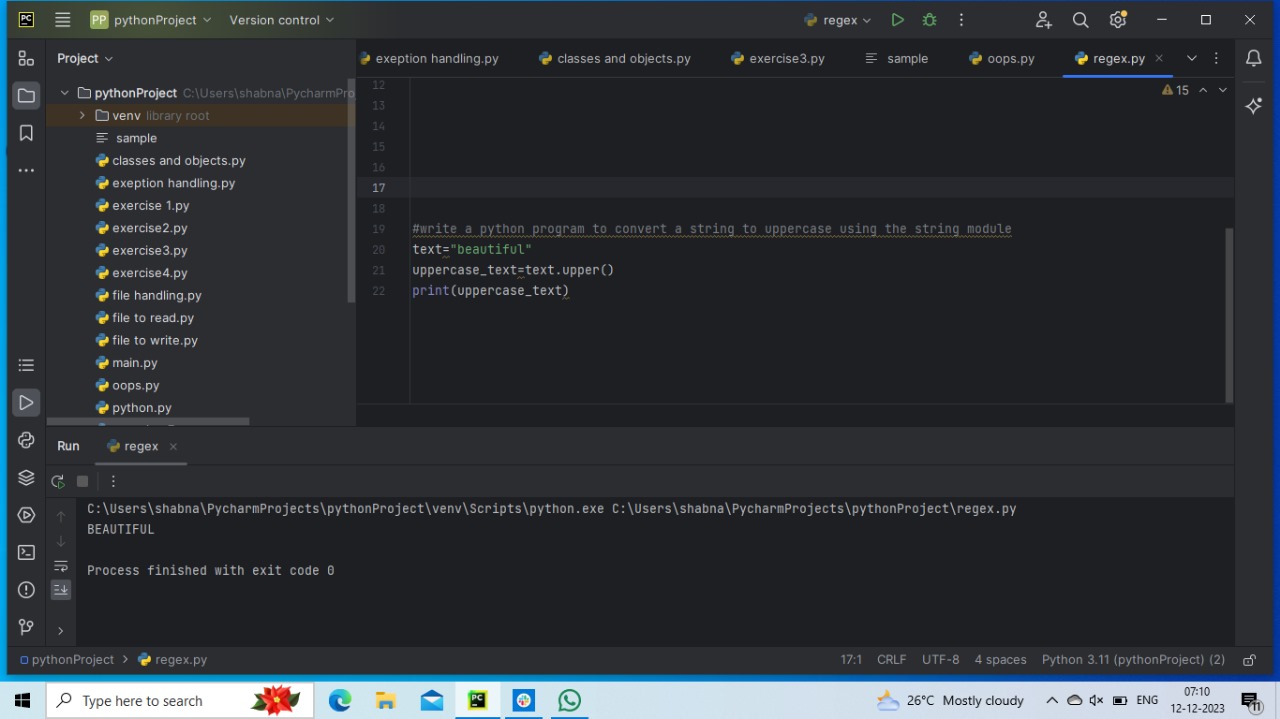


2.Write a Python program to generate a random number between 1 and 10 using

the random module.

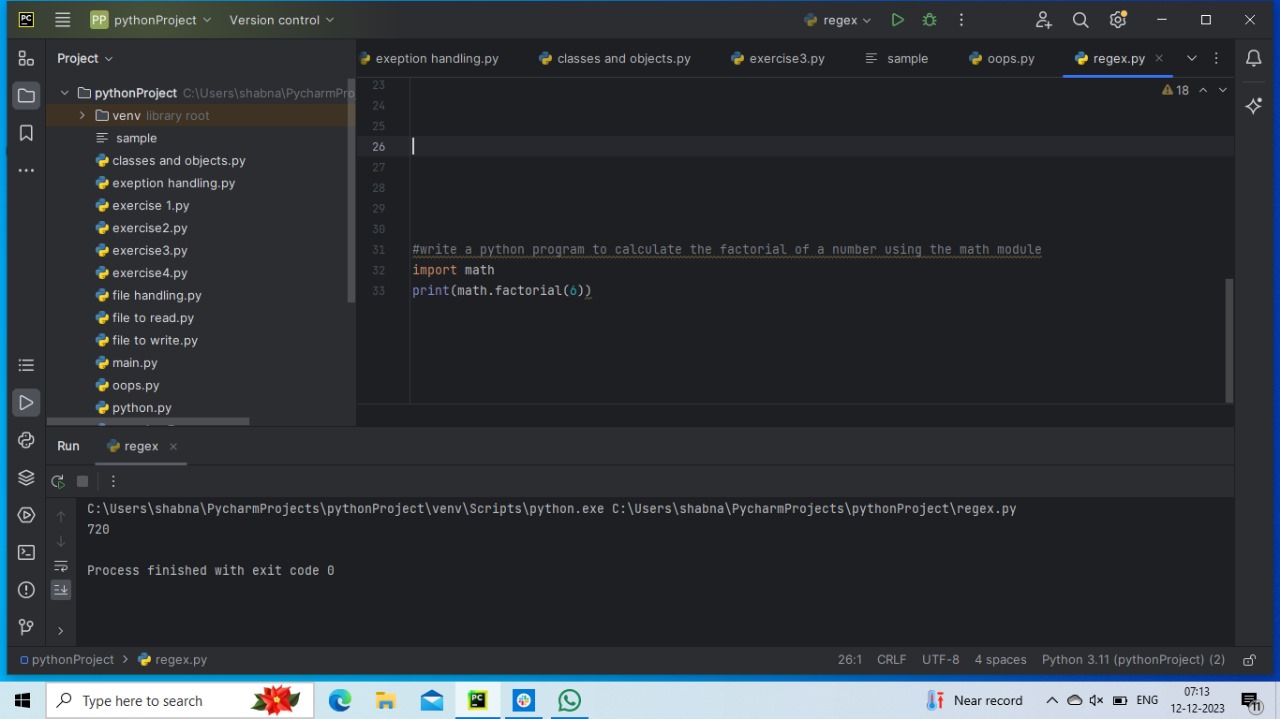
3.Write a Python program to convert a string to uppercase using the string

module.



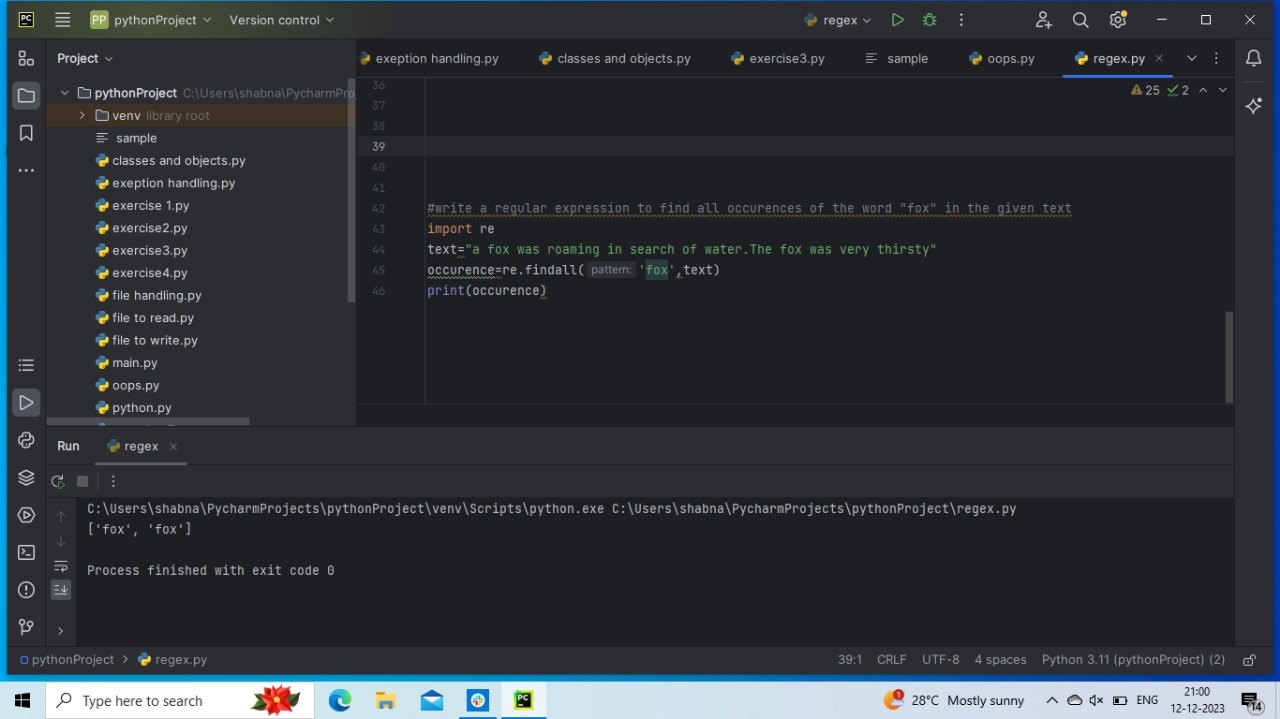
4.Write a Python program to calculate the factorial of a number using the math

module.



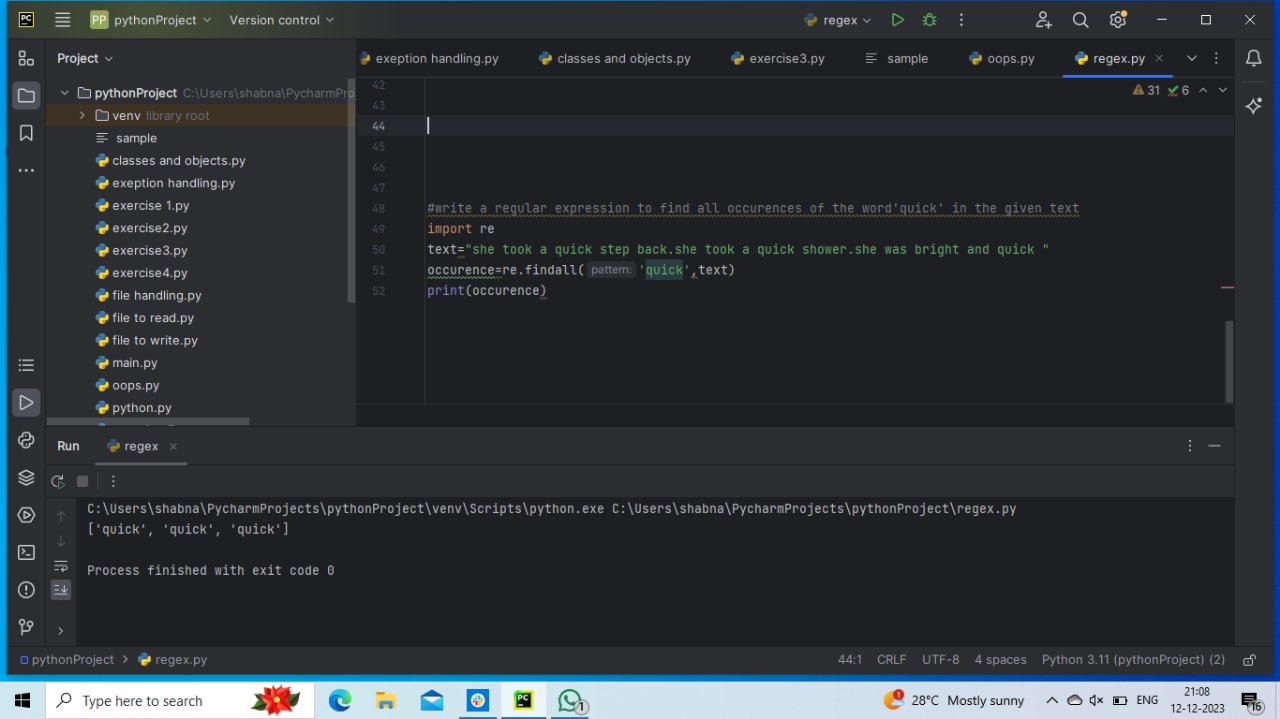
5.Write a regular expression to find all occurrences of the word "fox" in the given

text



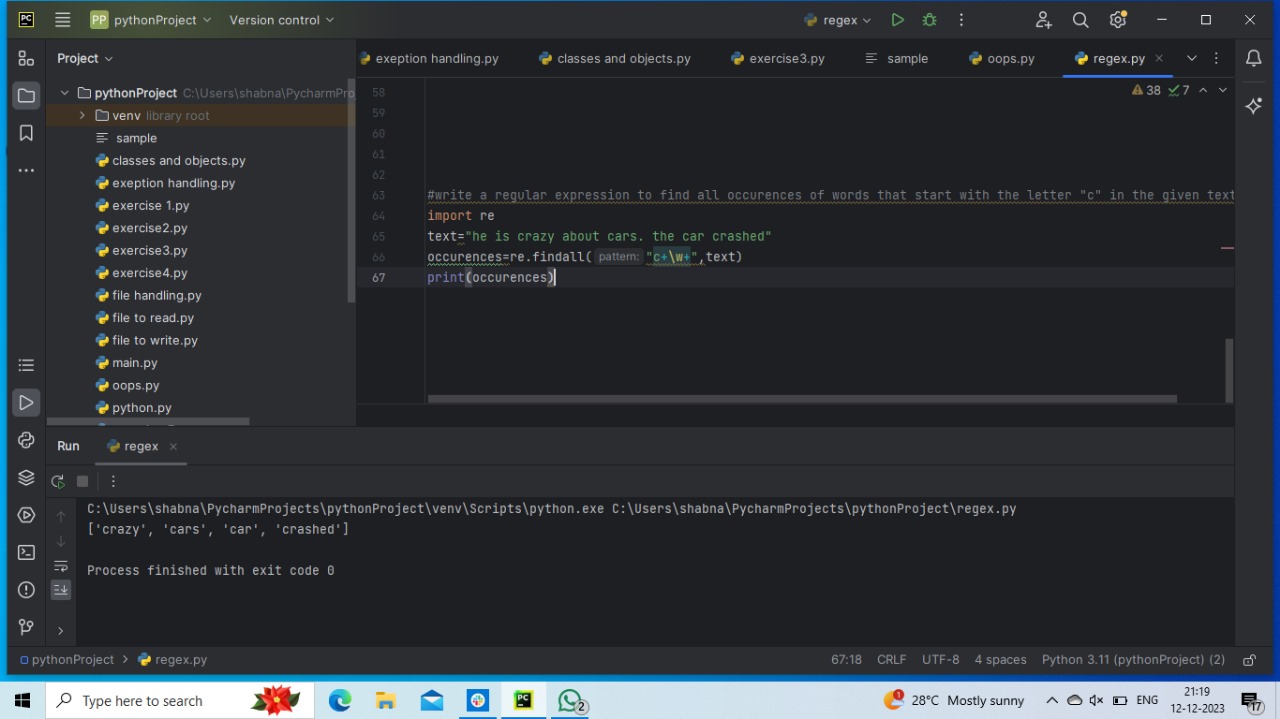
6.Write a regular expression to find all occurrences of the word "quick" in the

given text



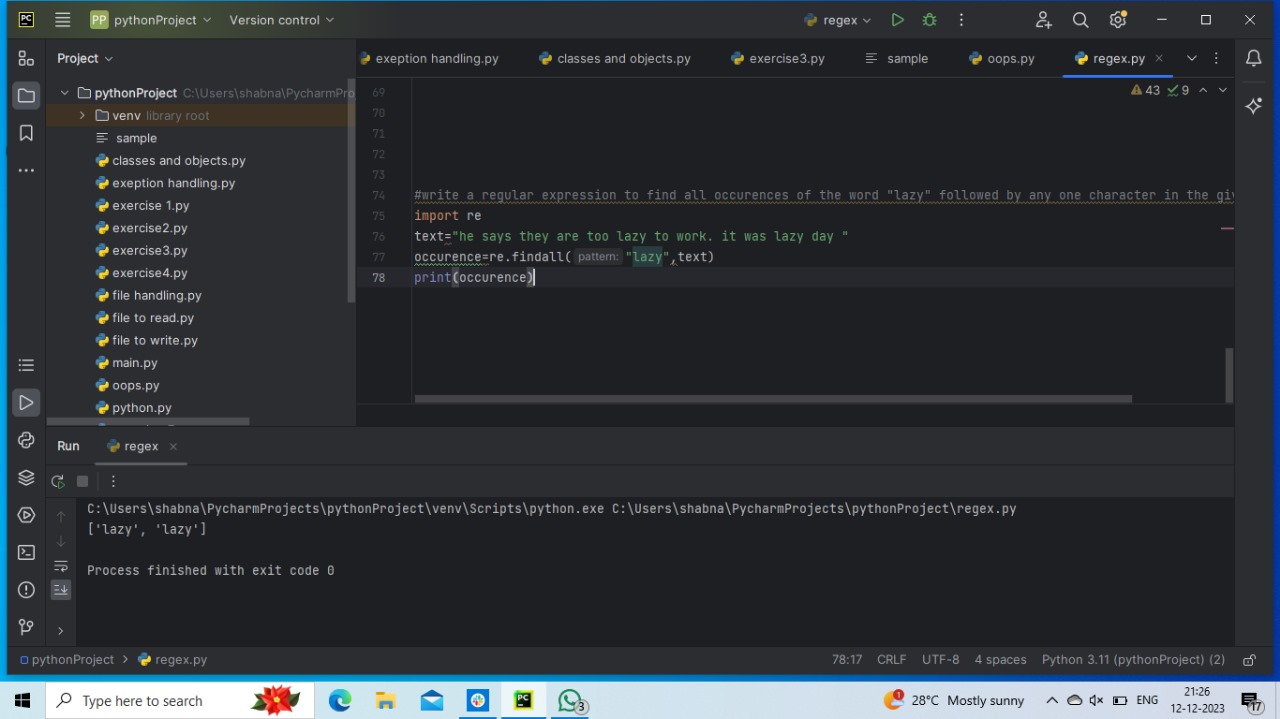
7.Write a regular expression to find all occurrences of words that start with the

letter "c" in the given text.

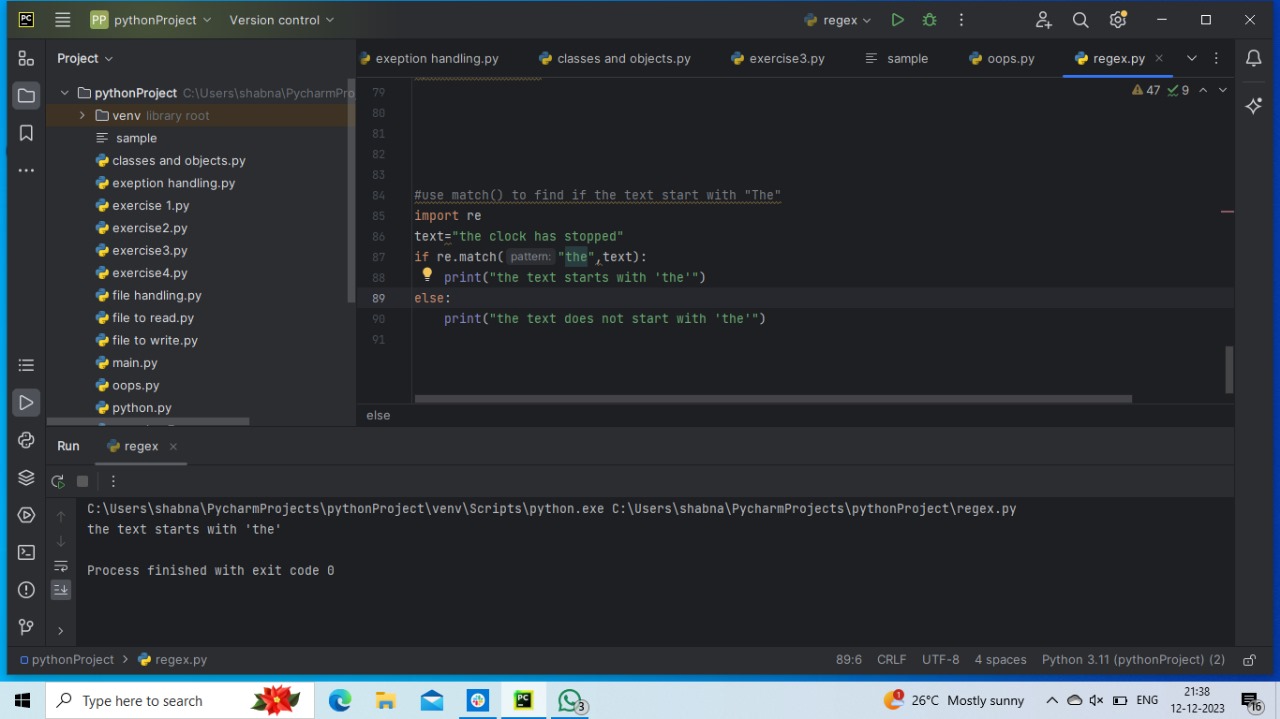


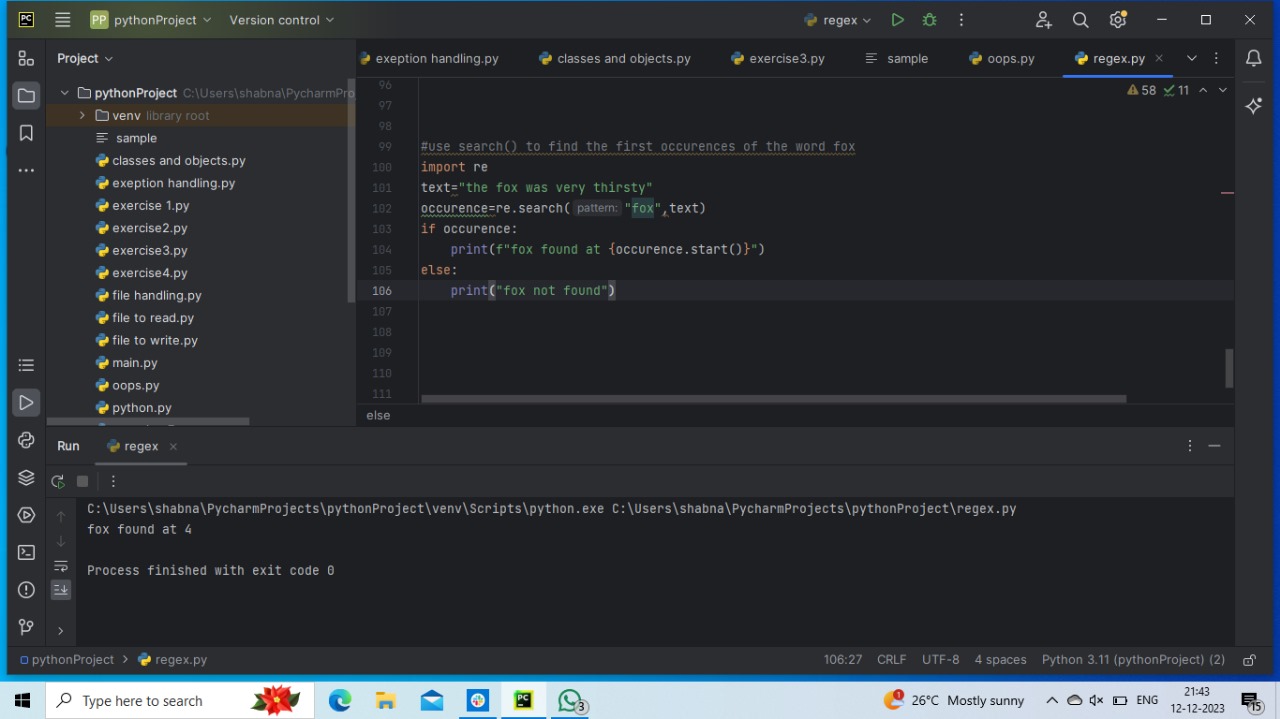
8.Write a regular expression to find all occurrences of the word "lazy" followed by

any one character in the given text.



9.Use match() to find if the text starts with "The”



10.Use search() to find the first occurrence of the word "fox”Numpy

1.Create a numpy array with 5 elements consisting of random integers between 1

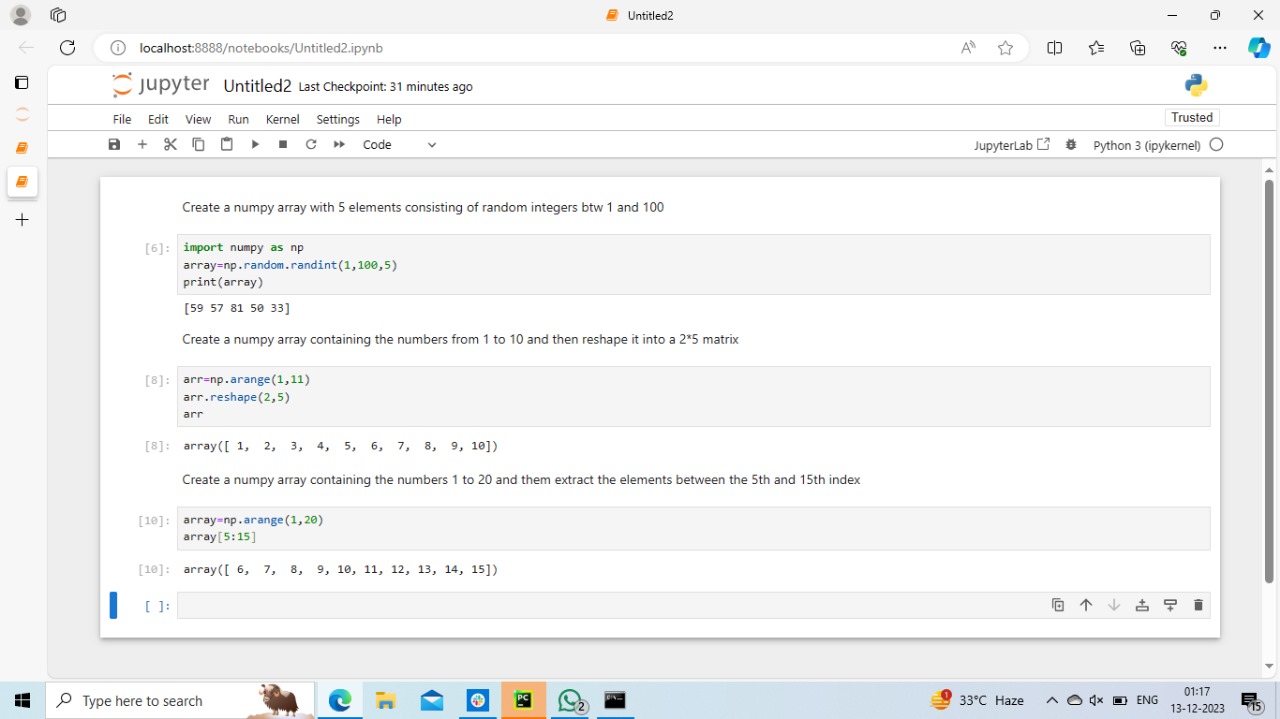
and 100.

2.Create a numpy array containing the numbers from 1 to 10, and then reshape it

to a 2x5 matrix.

3.Create a numpy array containing the numbers from 1 to 20, and then extract the

elements between the 5th and 15th index



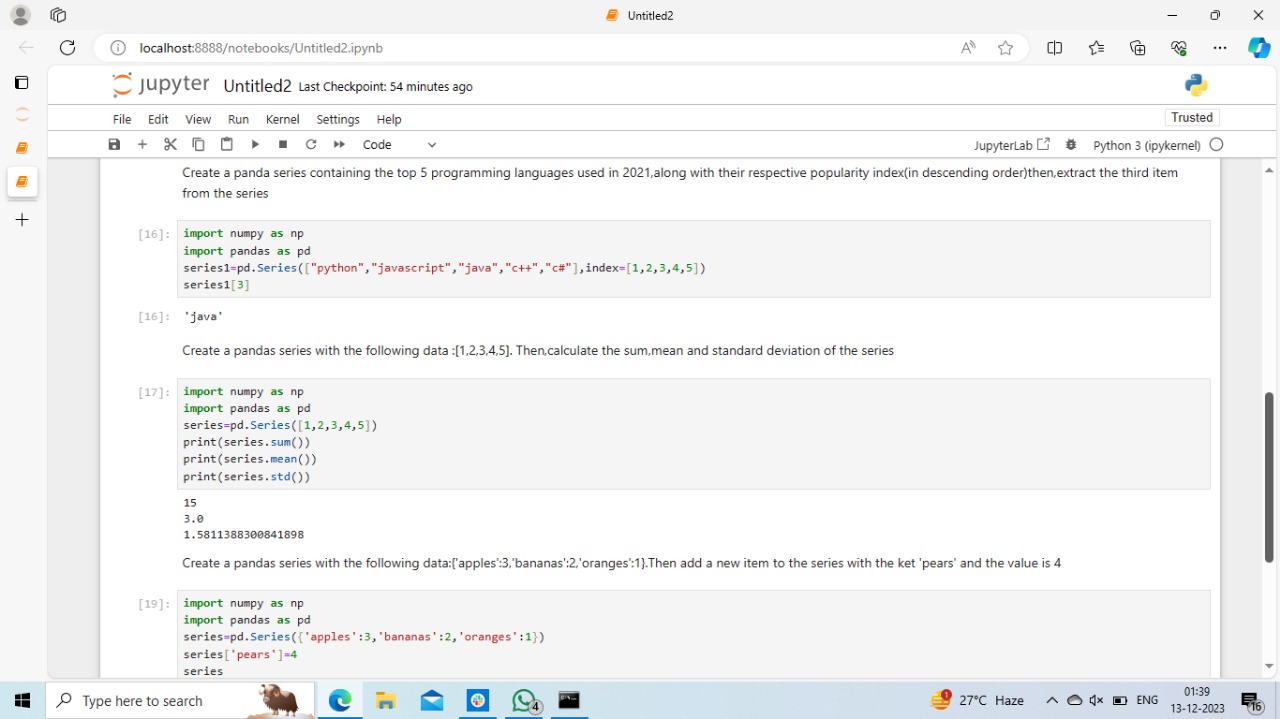
Pandas

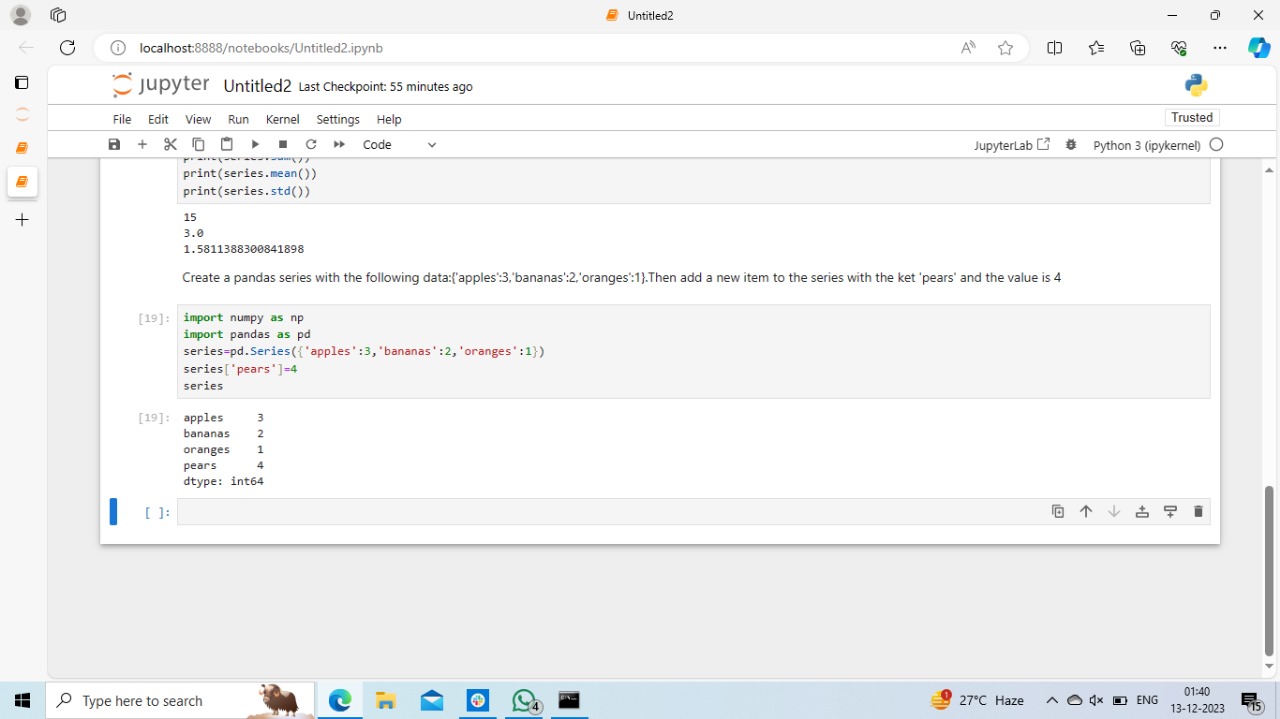
1.Create a pandas series containing the top 5 programming languages used in

2021, along with their respective popularity index (in descending order). Then,extract the third item from the series.

2.Create a Pandas series with the following data: [1, 2, 3, 4, 5]. Then, calculate the

sum, mean, and standard deviation of the series.





Dataframe

Exercise 1:

Create a dataframe with the following columns: name, age, and gender. The

dataframe should have 10 rows of data, with the following values

Exercise 2:

Add a new column to the data frame created in question 1, called occupation. The

values for this column should be Programmer, Manager, and Analyst,

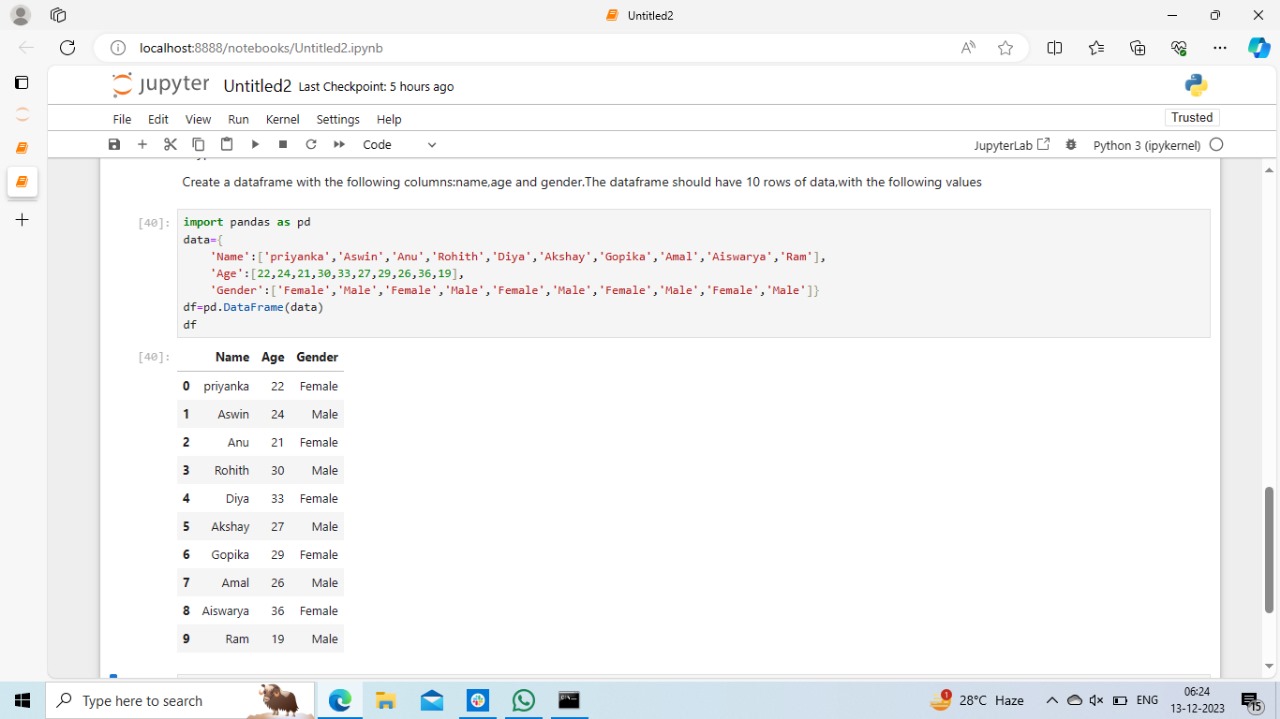
corresponding to the rows in the dataframe

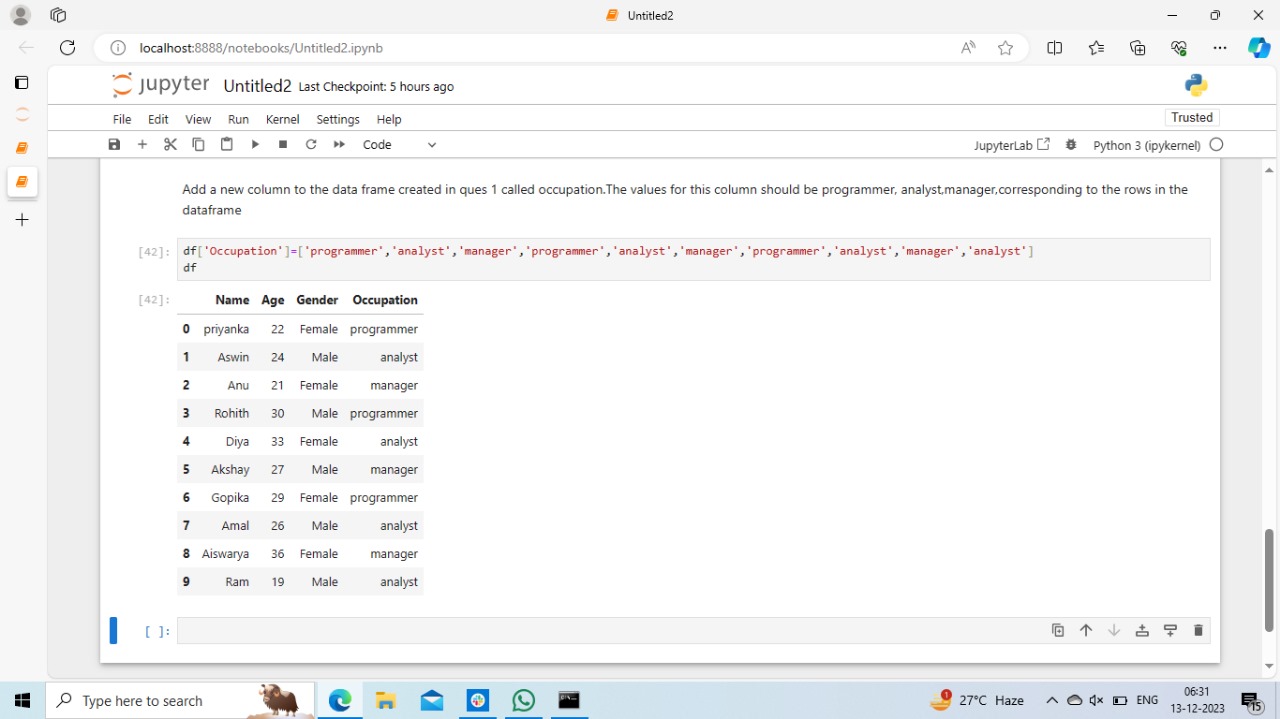
Exercise 3:

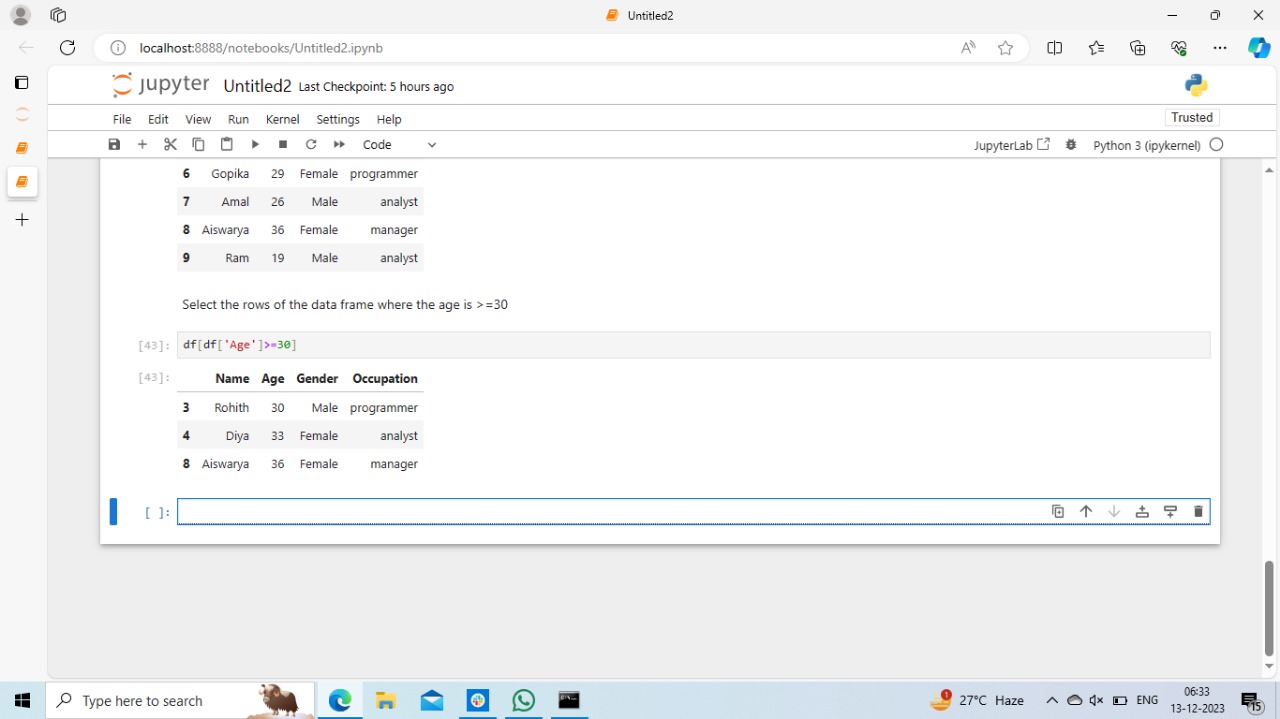
Select the rows of the dataframe where the age is greater than or equal to 30.

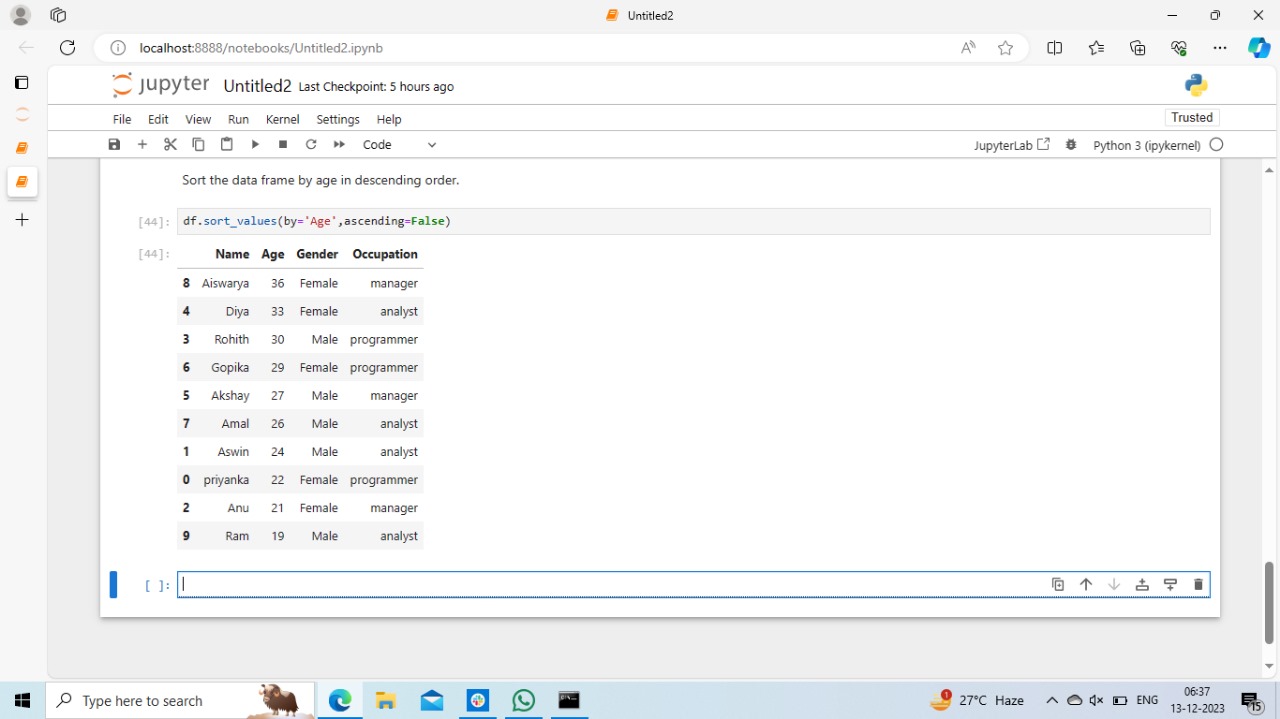
Exercise 4:

Sort the data frame by age in descending order.









Matplotlib:

Exercise 1:

Create a line plot using matplotlib pyplot that displays the population of four

different cities over time. Each city should have its own line, and the x-axis should

represent years (e.g. 2010, 2011, 2012, etc.) while the y-axis should represent the

population.

The data for the four cities is provided below:

City A: [500000, 550000, 600000, 650000, 700000, 750000, 800000]

City B: [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]

City C: [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]

City D: [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]

Exercise 2:

Create a scatter plot using matplotlib pyplot that shows the relationship between

the number of hours studied and the test scores obtained by a group of students.

Use the following data:

Hours Studied: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Test Scores: [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

Exercise 3:

Create a bar chart using matplotlib pyplot that shows the total sales for each

month of the year. Use the following data:

Month: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct",

"Nov", "Dec"]

Sales: [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346,

6675]