**Python coding Assessment : 20-12-23 Abhiram Basa**

1. **Explain Python Module with examples**

A Python Module is a file containing Python definitions and statements.

Modules are the files which we can create and import and use them in other python code files.

There are 2 types of modules in Python .

* Pre defined modules.
* User defined modules.

**Pre defined modules:** These are already developed by the python developers in the open source, we can use them by importing them.

Examples: math, random, etc.

**User defined modules:** These are created or defined by the users.

**a.Import module in Python :**

**Syntax:** import Module\_name

**Example: import math**

**b.Renaming the Python module:**

In order to rename a python module, we need to use “ Alias “.

The keyword which we use is **“ as “** while importing the module.

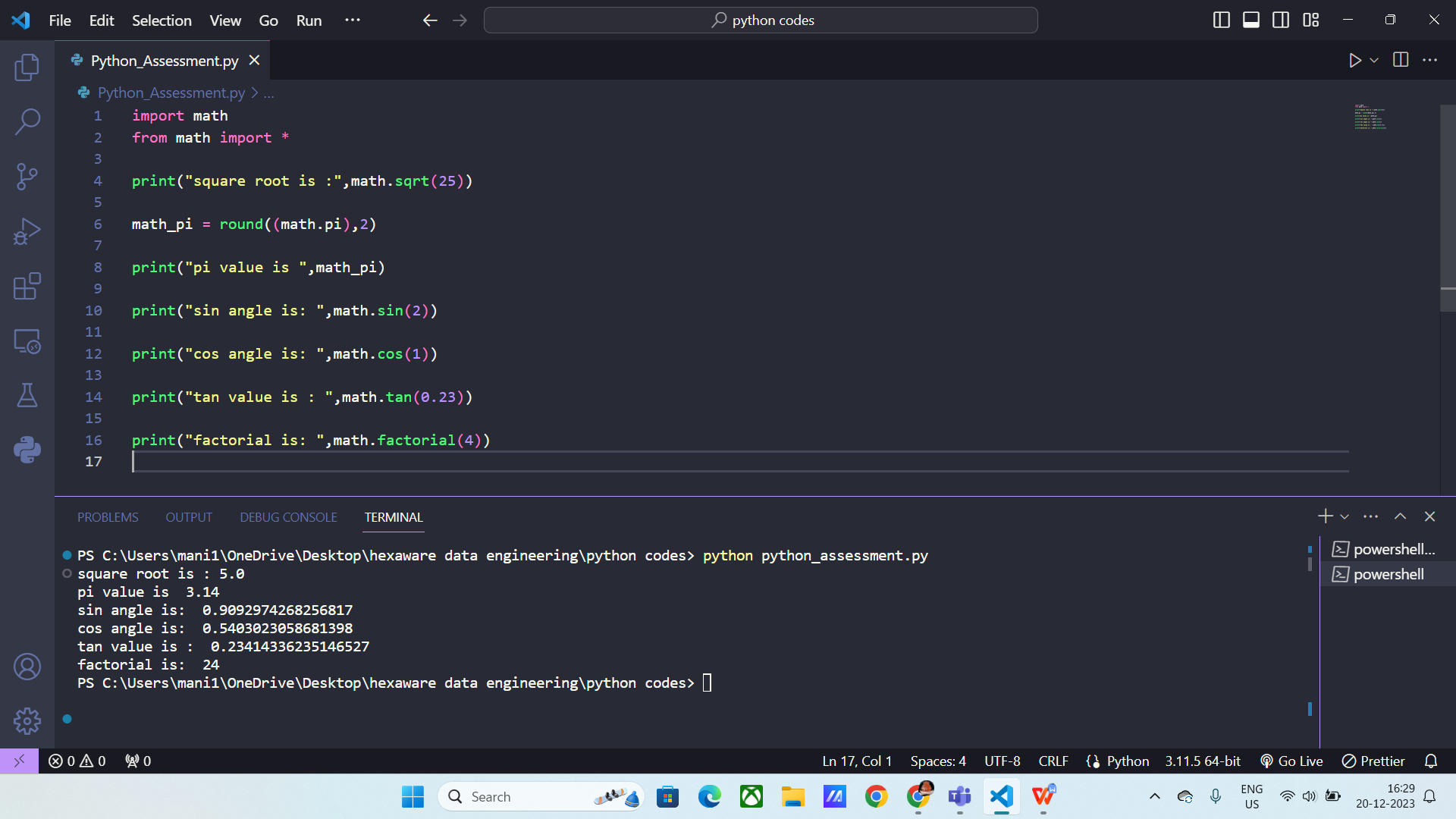
**Syntax:** import module as Alias\_name

**Example: Import pandas as pd.**

Here pd acts as Alias name for pandas module.

Below I imported math module and did hands on practice.

**Hands on Practice:**



In the above code I used like

**square root(sqrt):**  gives the square root of the given num.

Ex: sqrt(36) , output: 6

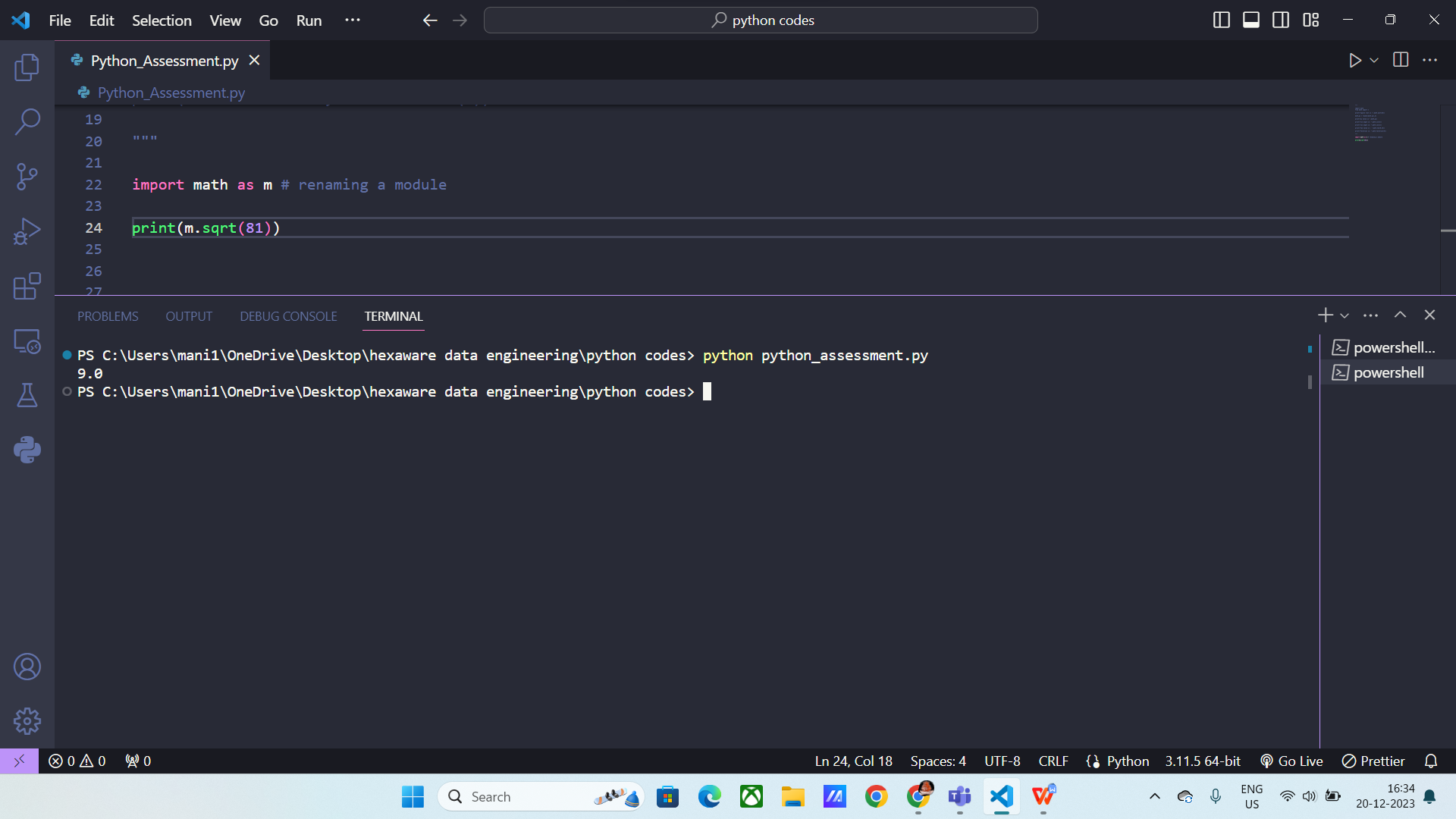
**Math.pi :** Returns the “pi” value.

**Sin,cos,tan :** Returns the sin,cos,tan values.

**Factorial:** Return the factorial of a number

Ex: 5! output = 5\*4\*3\*2\*1 = 120

Below I renamed math module as m and used it to find the square root of the number.



**User defined modules hands on practice :**

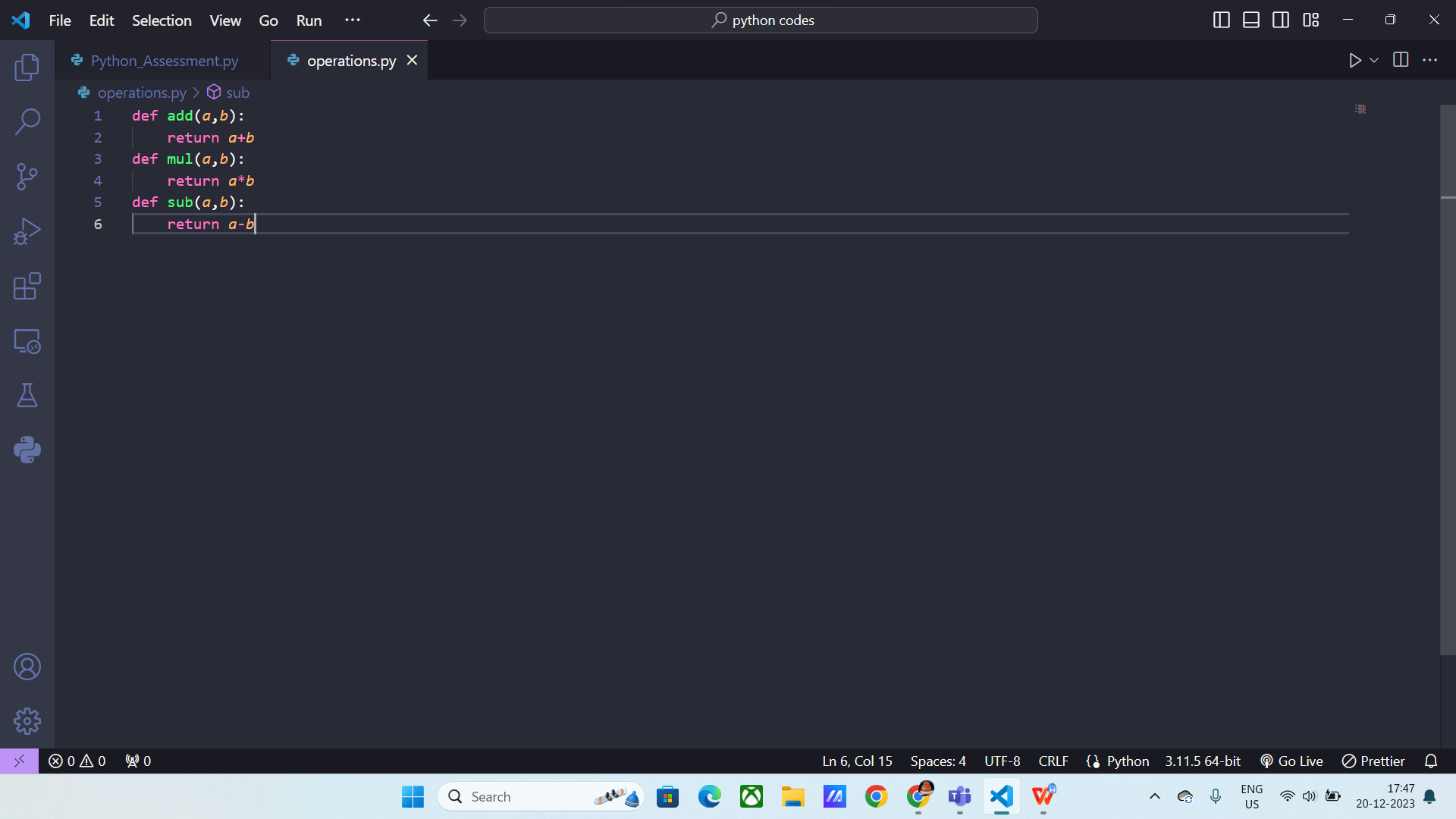
* First we create a file called called “operations”.
* It contains addition, substraction, multiplication operations, now we can import that into mail code file by using
  + Import filename
  + Ex: import operations.
* Now we can call methods inside the operations file in the main file using the **filename.operation\_name.**

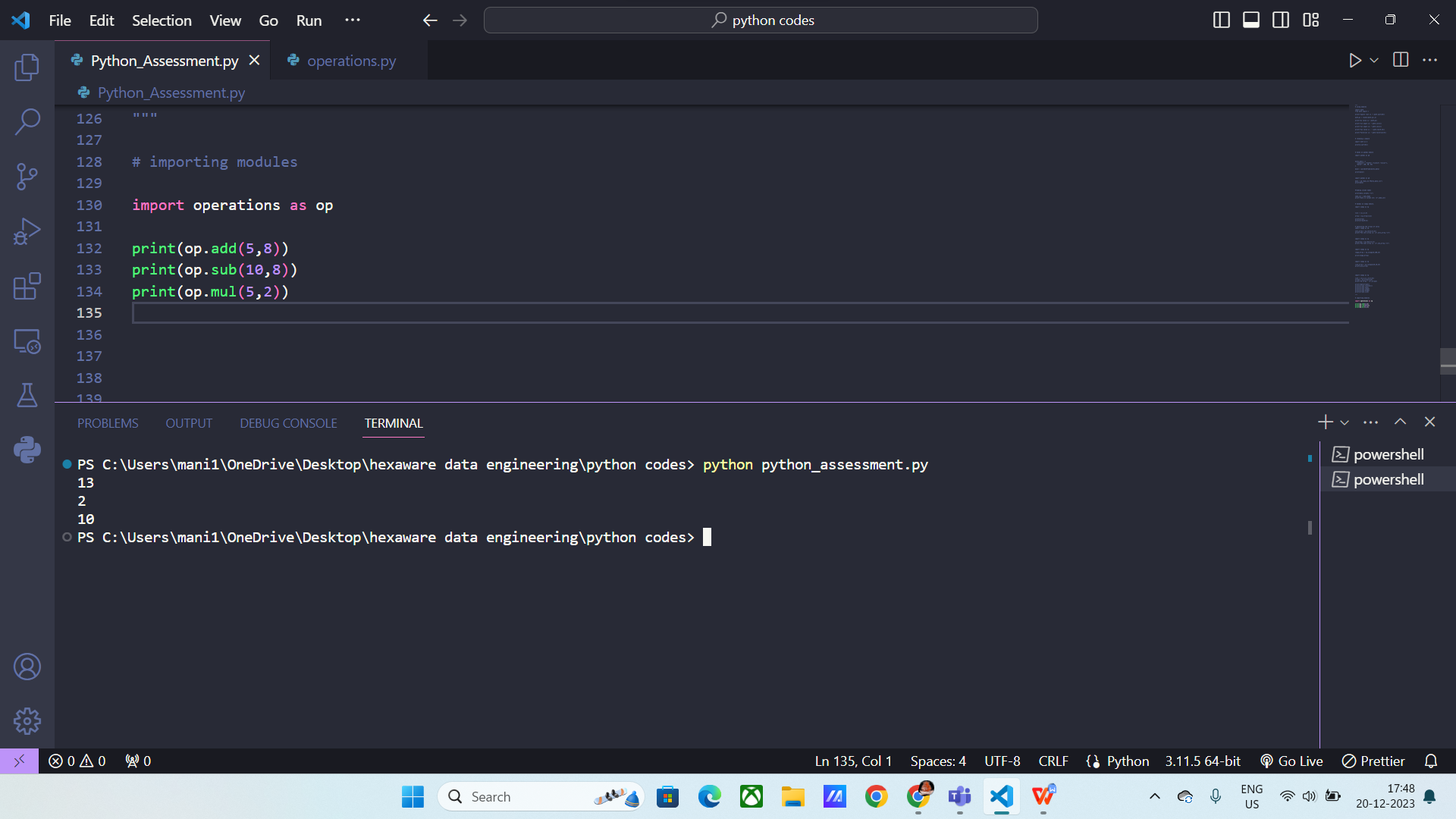
Ex: operations.add(5,8)

operations.sub(5,8)

operations.mul(5,8)

Here is the operations file:





1. **Explain Pandas and numpy using Examples in PYTHON**

**Pandas:** Pandas is a python library used for data analysis. It is used to perform operations on files.

In order to perform any operations using pandas, we need to import it.

**Syntax:** import pandas as pd

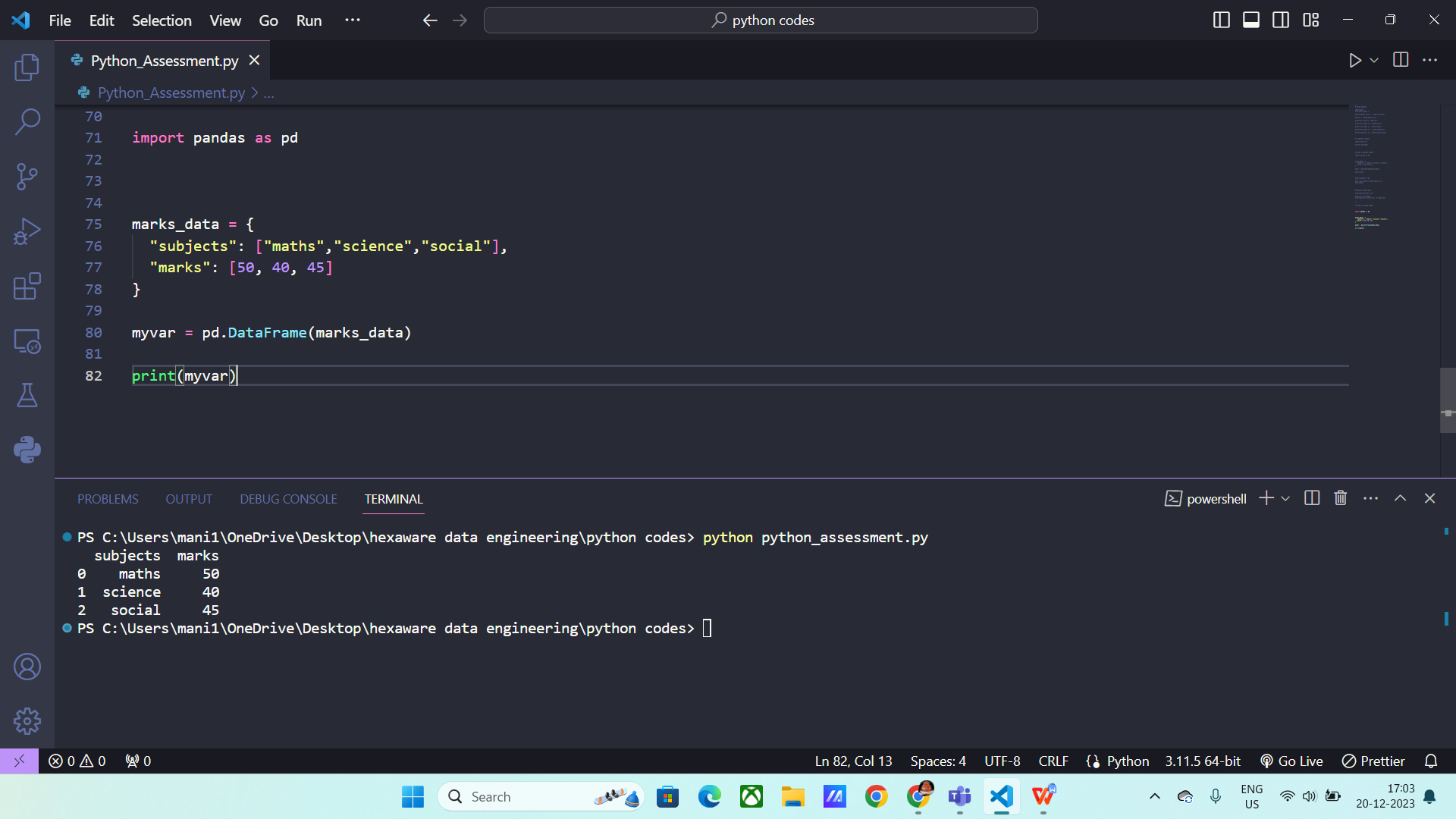
Using pandas module we can analyse huge amount of data with its built in methods.

Below I used built in method called Data frame.

**Data frame:** This method is used to convert the data into rows and columns format.

Below you can see , the marks\_data is converted into data frame.

**Syntax: pd.DataFrame(marks\_data)**

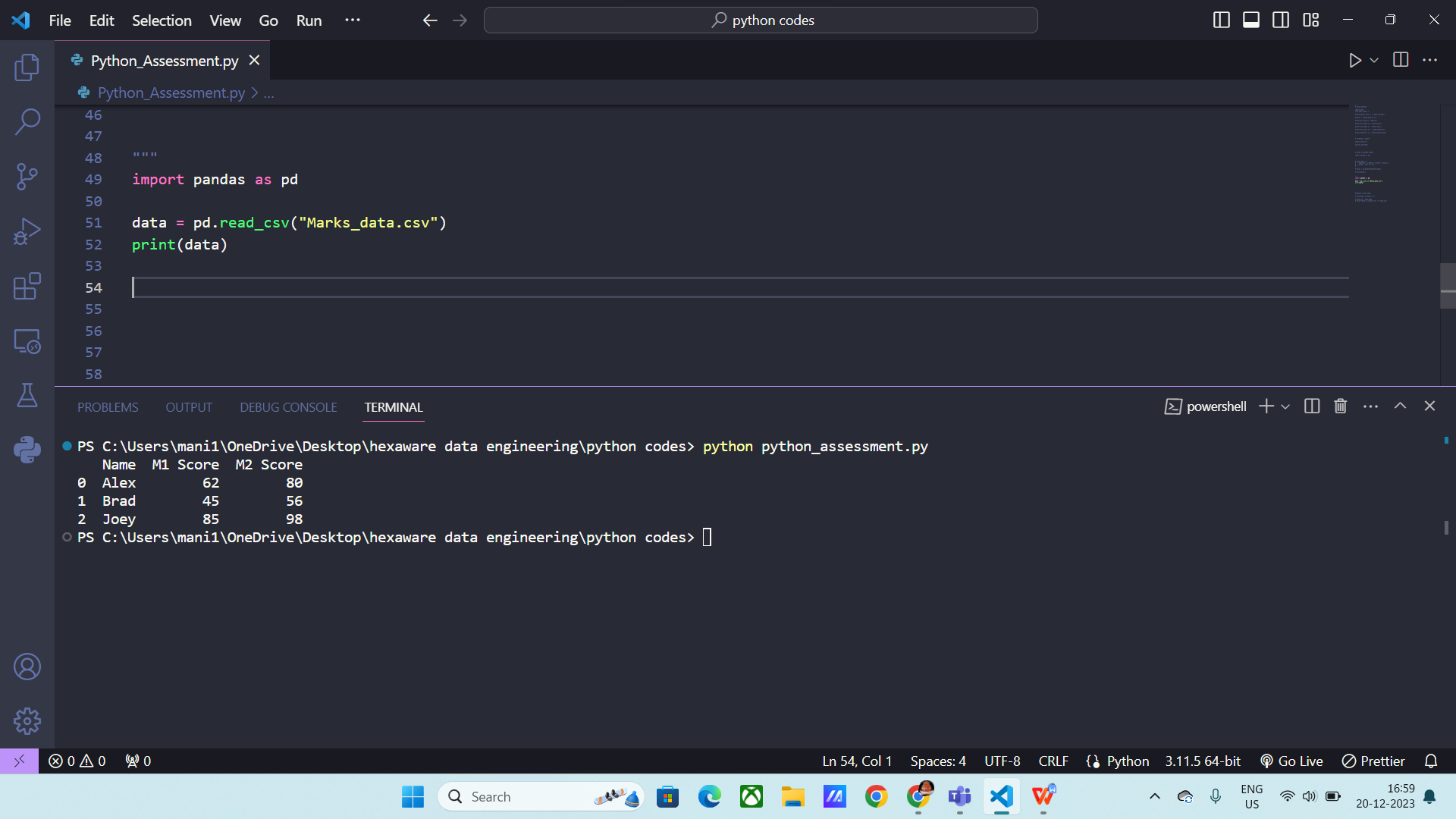


Next I’ve took a csv file and performed some operations on it using pandas module.

**Read() using pandas:**

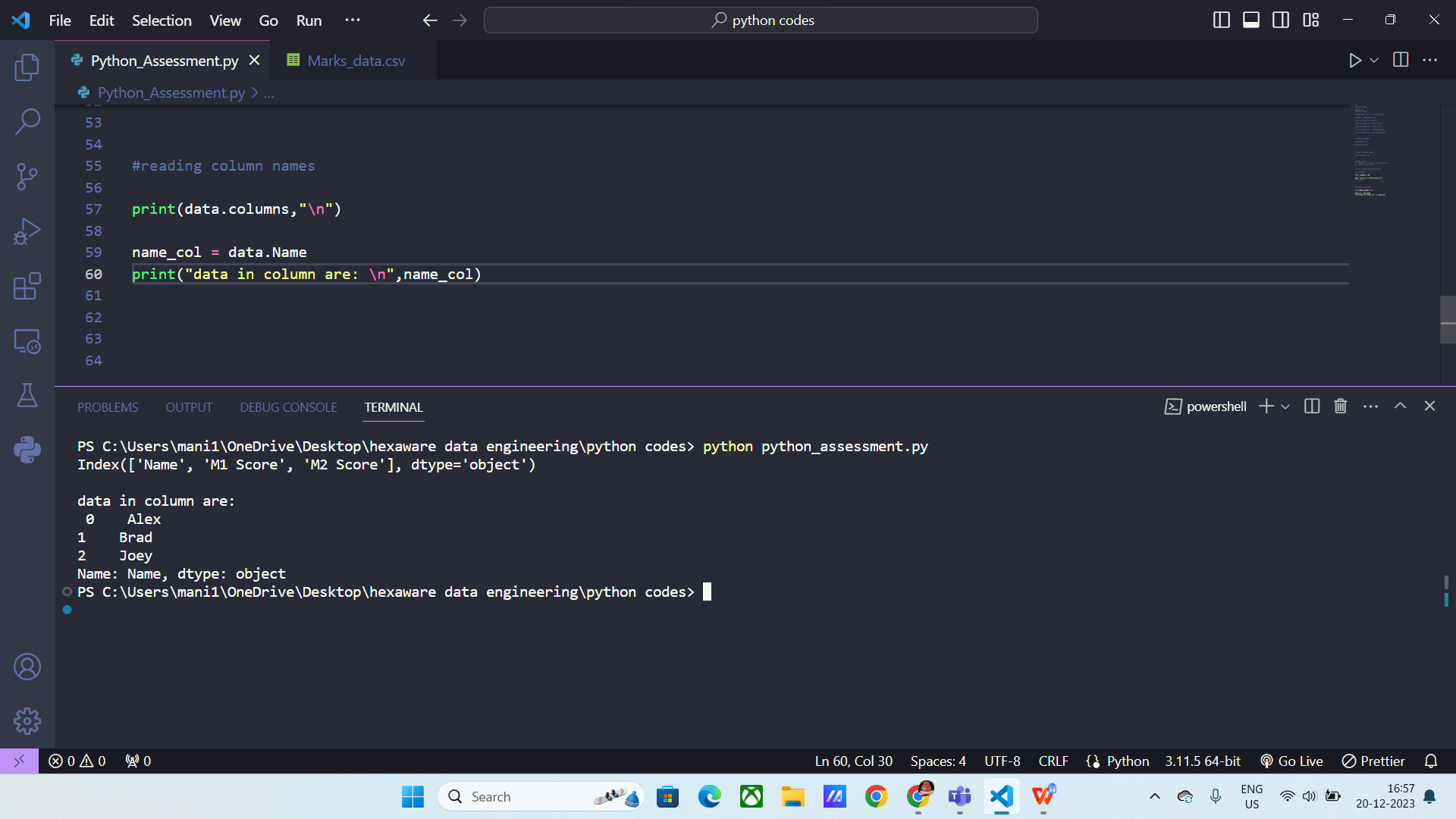
Syntax: pd.read\_csv(“file name”)

It will print all the data in the marks details.



**Printing columns:**

We use (.columns) to return all the columns names.



**Numpy module:**

Numpy is mainly used to work with arrays in python. It is one of the python module also known as “Numerical Python “.

Numpy module can work with 1D arrays, 2D arrays and 3D arrays.

**Syntax: import Numpy as np**

In order to work with data using numpy we need to convert the data into Array form.

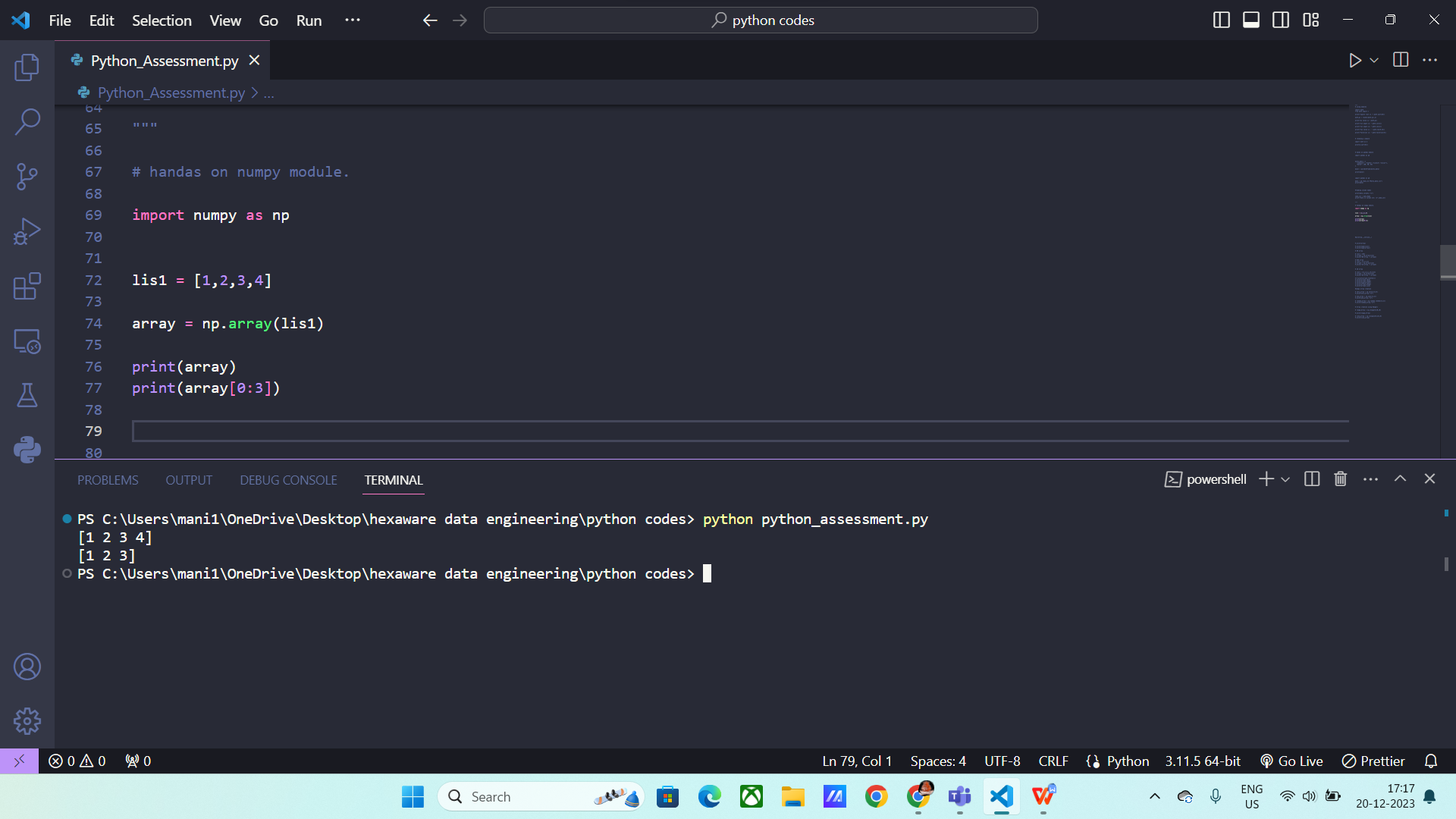
As numpy works with Arrays.

There are four ways to create Arrays

* **Array() :** This method is used to convert data into arrays.

Syntax: np.array([data])

You can convert list, tuples, etc. Into arrays using this method.



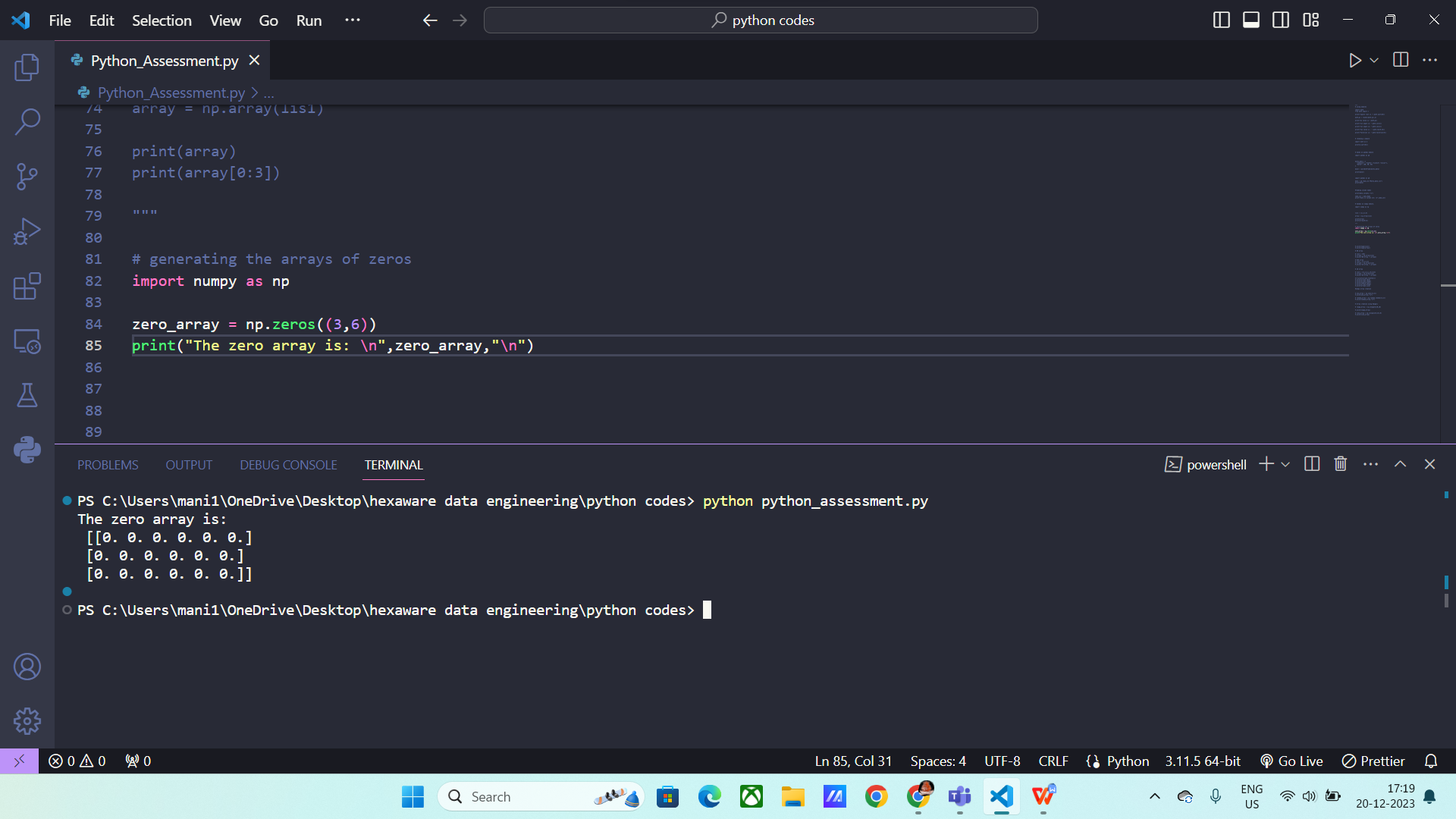
Above I used array() method to convert into arrays form.

* **Built-in methods:**

There are few built in methods to generate an array.

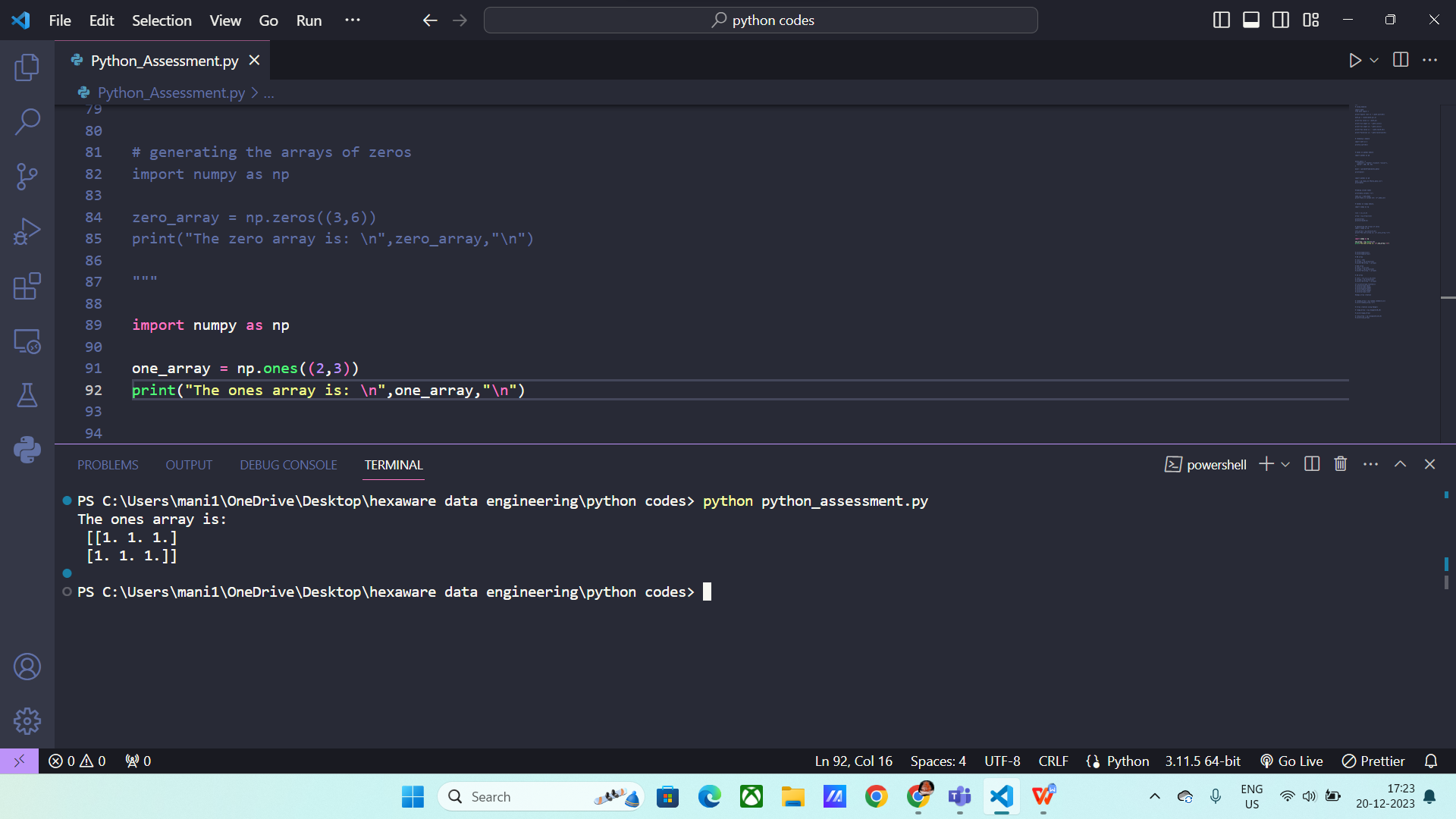
* **Zeros():** It will generate the array with all zeros inside it.

**Syntax:** np.zeros((row,column))



* **Ones() :** It will generate the entire array with ones.

**Syntax:** np.ones((row,column))

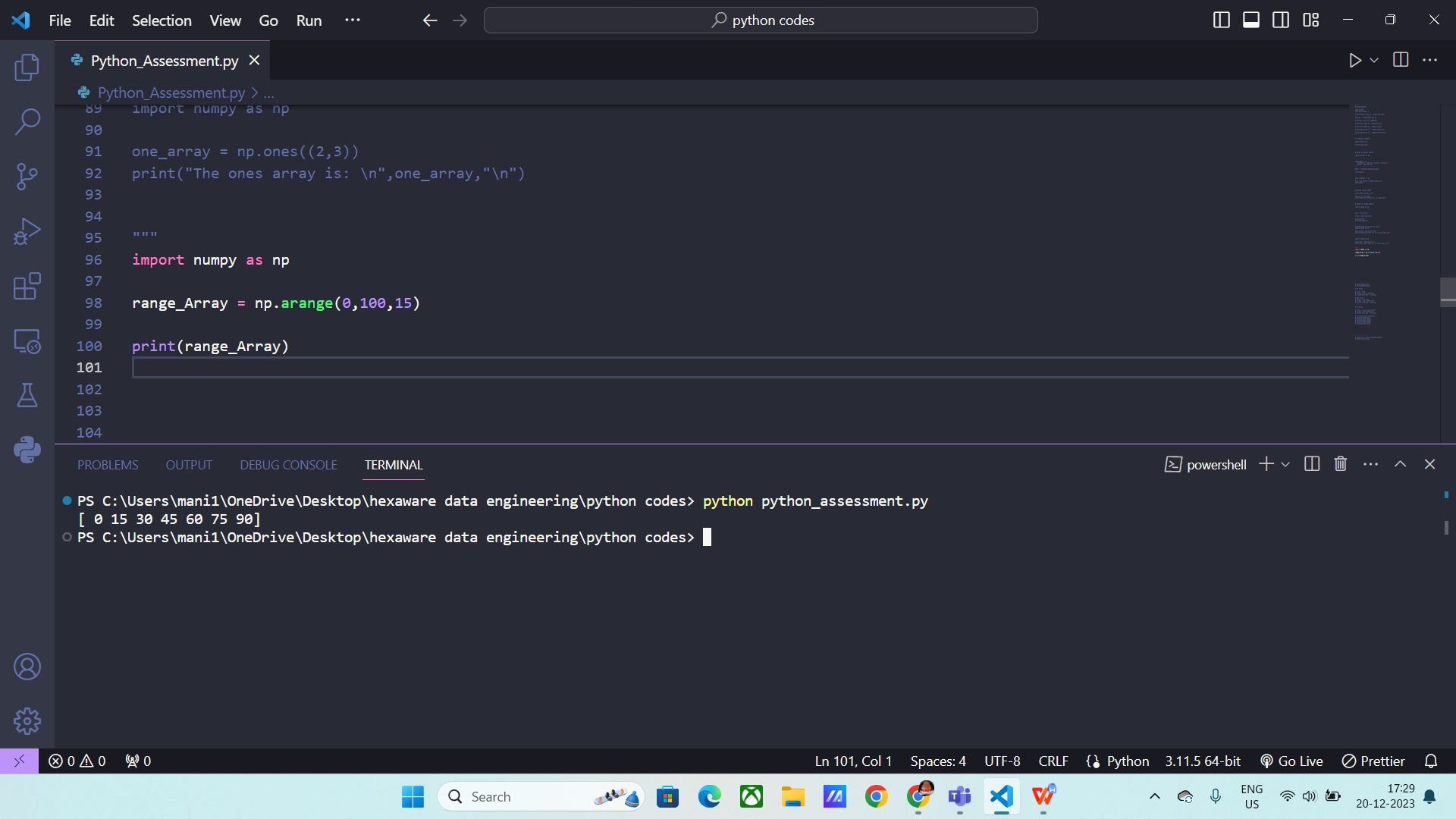


Arange() : This method will create array based on specific condition.

Syntax: np.arange(start,stop, step)

Ex: np.arange(0,100,15)

Here it will create an array from 0 to 100 with a step of 15.

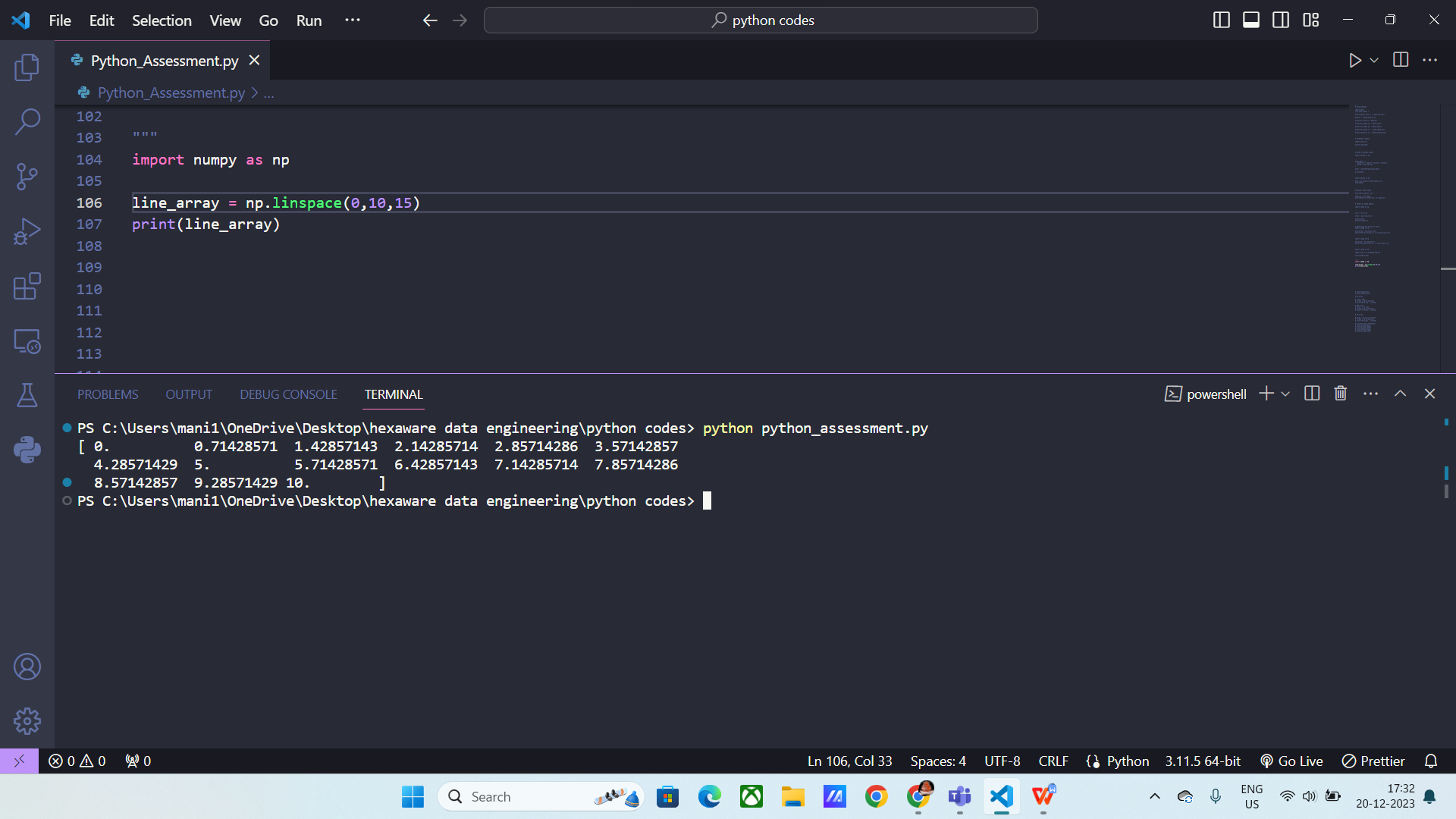


The last method to create an array is

* **line space():** It will similar as arange() but instead of step , you need to give how many values to print.

Syntax: np.linespace(start,stop, how\_many\_values\_you\_want)

Ex: np.linespace(0,10,5)



Methods:

**Type(array)**: It will give you the type of array I.e “nd.array”

**Array.flatten()**: It will flatten the array.

**Array.ndim:** It will returns the dimension of the array.

**Array.size:** It will return the total no of elements

**Array.shape:** It will return the no.of rows and columns the array has.

