**Day- 14**

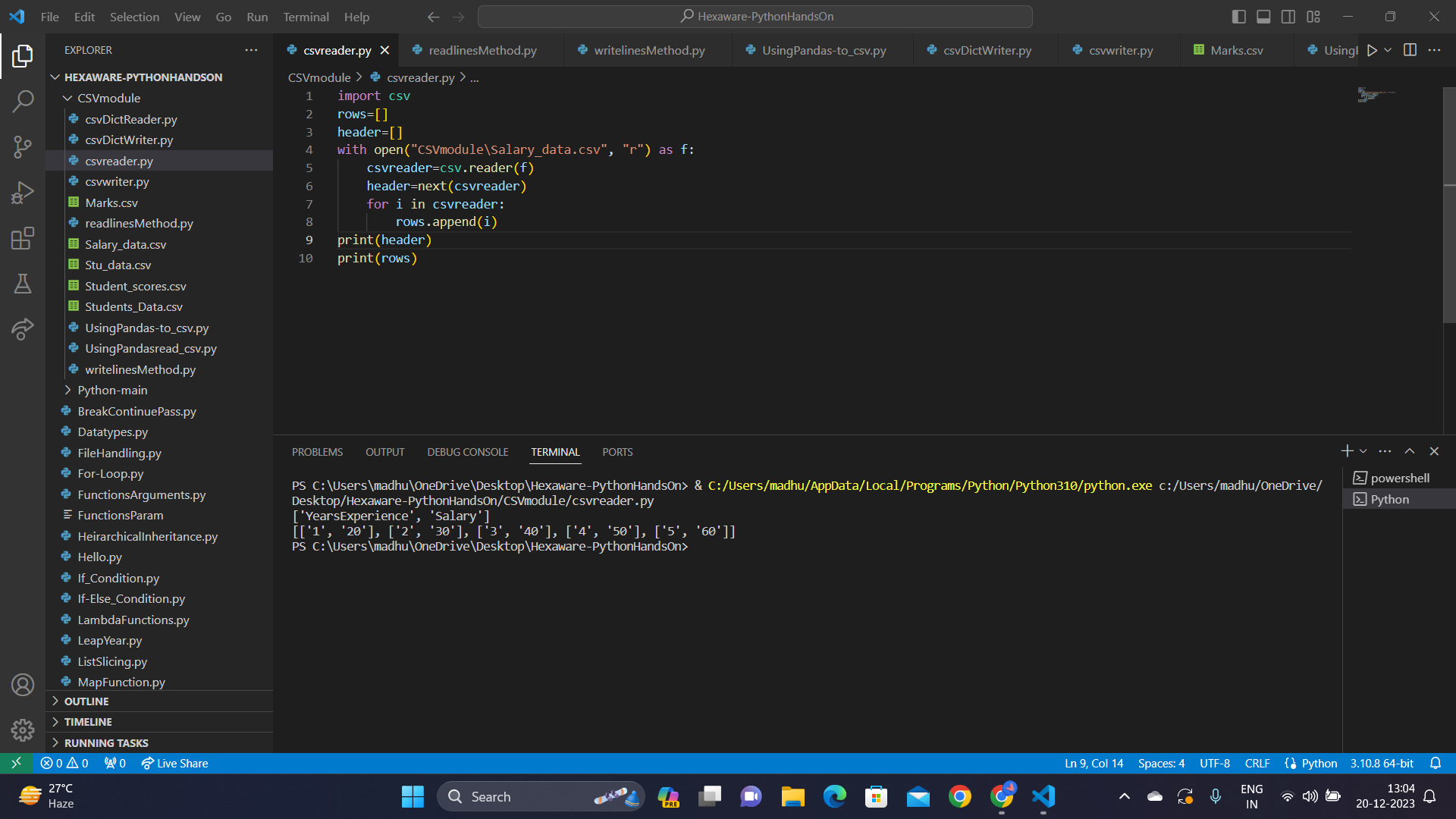
**Madhu Kalyani Gadi (19-12-2023)**

CSV is a simple text file format for organizing tabular data, widely used in data science. Each line in a CSV file represents a record, and fields within the record are separated by commas.

There are 4 ways to read csv files in python.

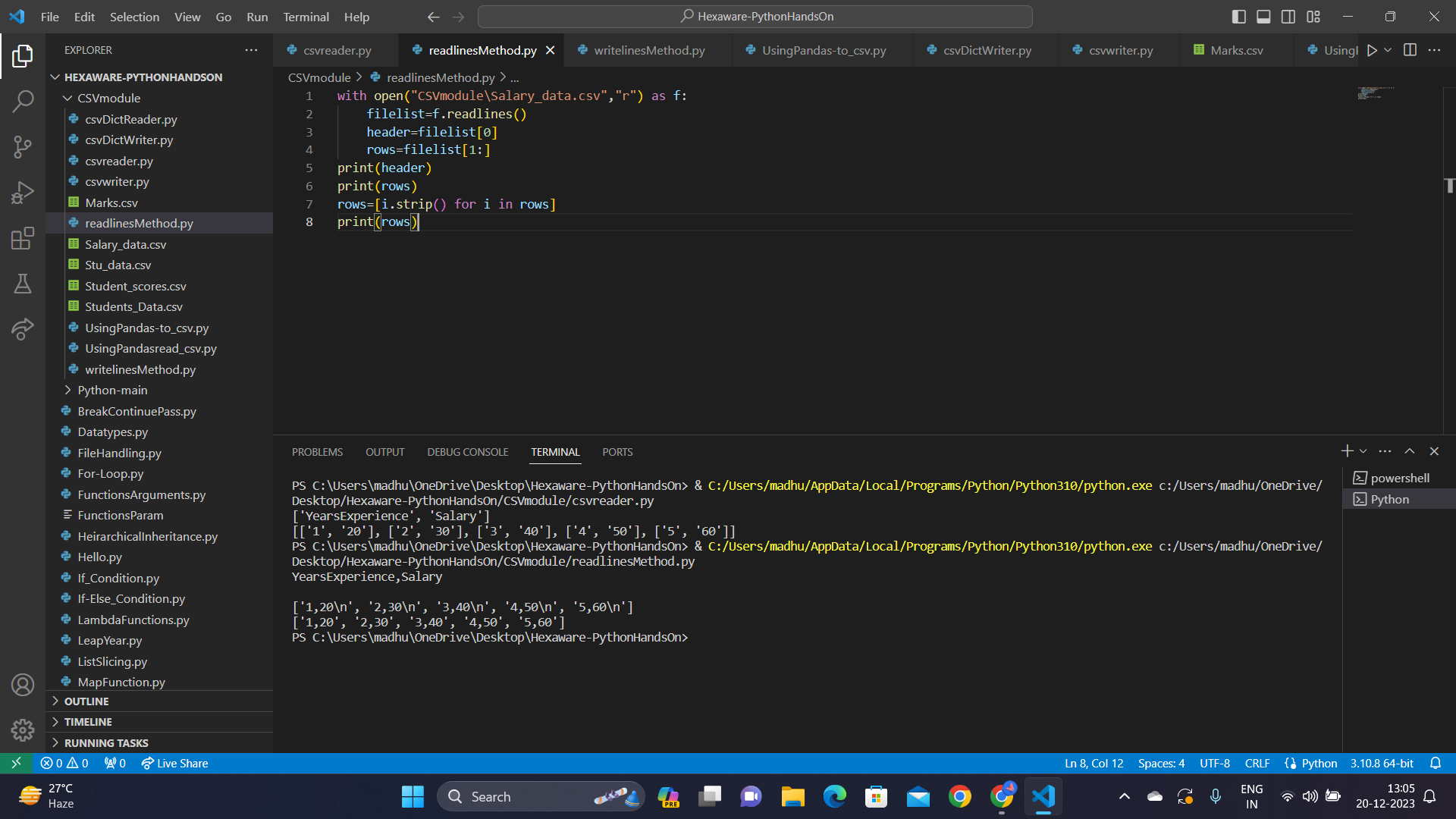
1. Using csv.reader()

csv.reader() in Python, you can read a CSV file by creating a reader object, which helps you iterate through the file's lines and easily extract data fields.



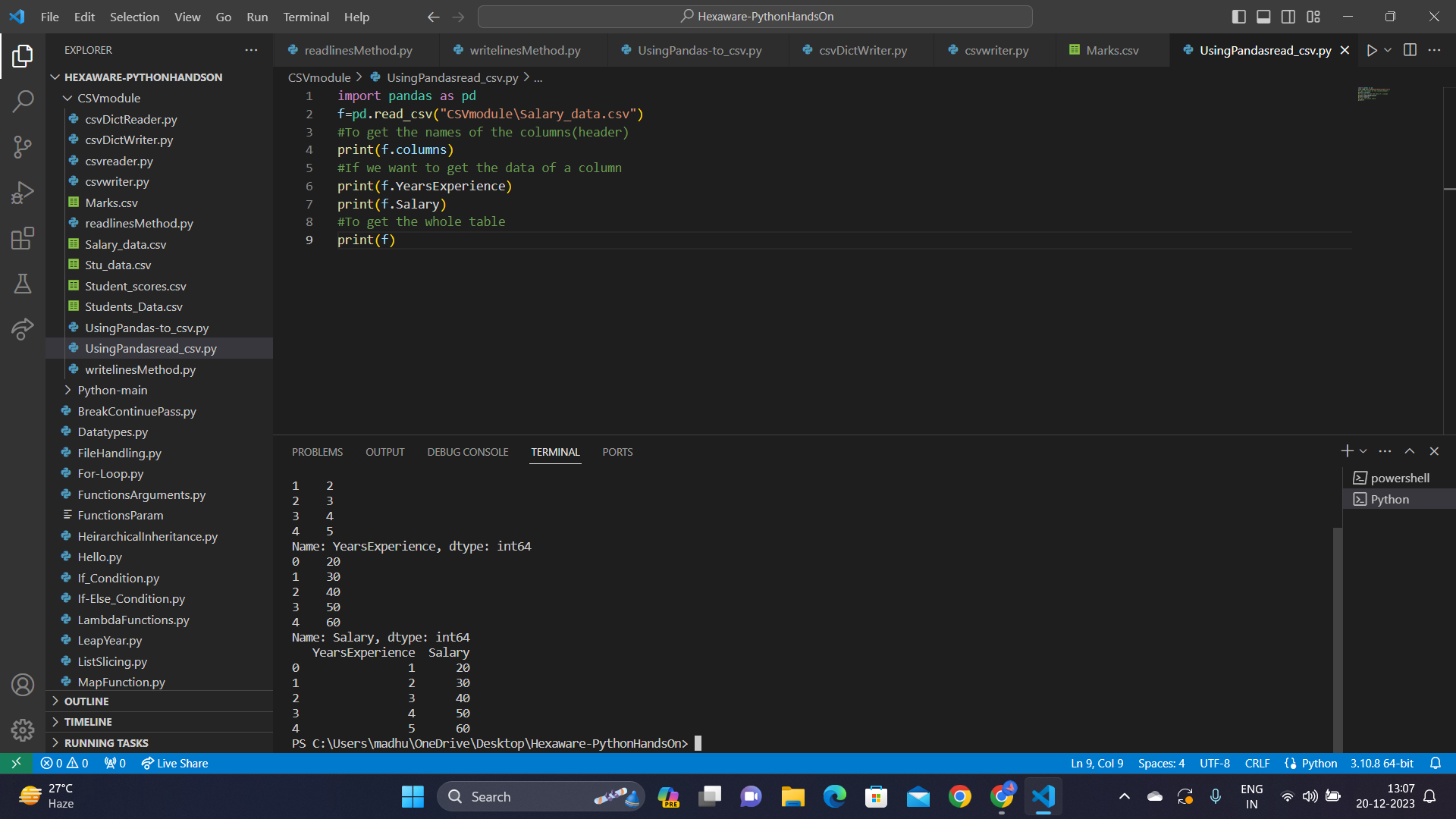
2. Using .readlines() function

With .readlines() in Python, we can read a CSV file, and it gives you a list where each element represents a line. It's like creating a to-do list for your program, making it easier to work with the file's content.



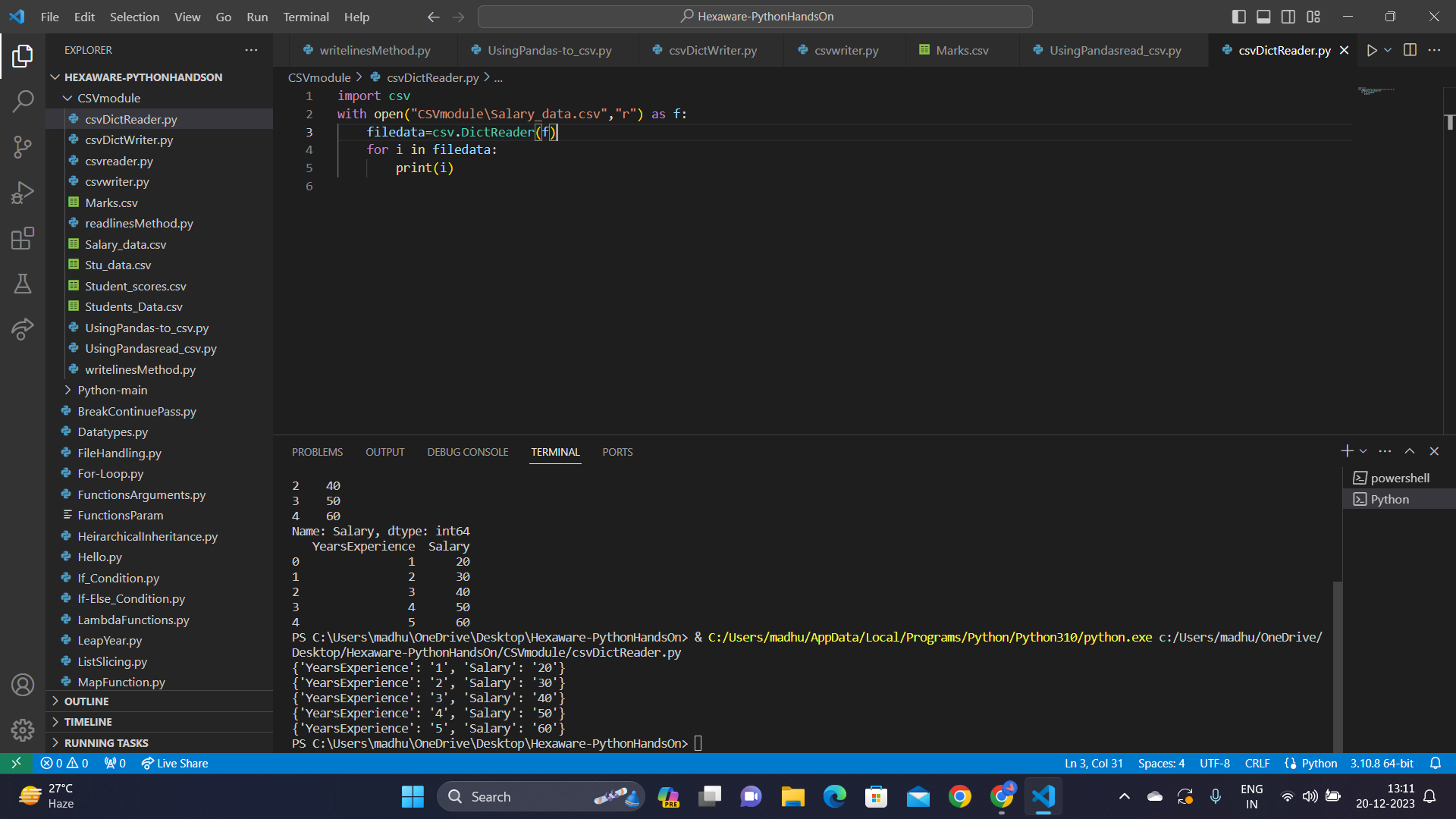
3. Using Pandas

Pandas in Python is like using a powerful tool that directly converts CSV data into a structured table (DataFrame), simplifying data analysis and manipulation tasks with ease.



4. Using csv.DictReader

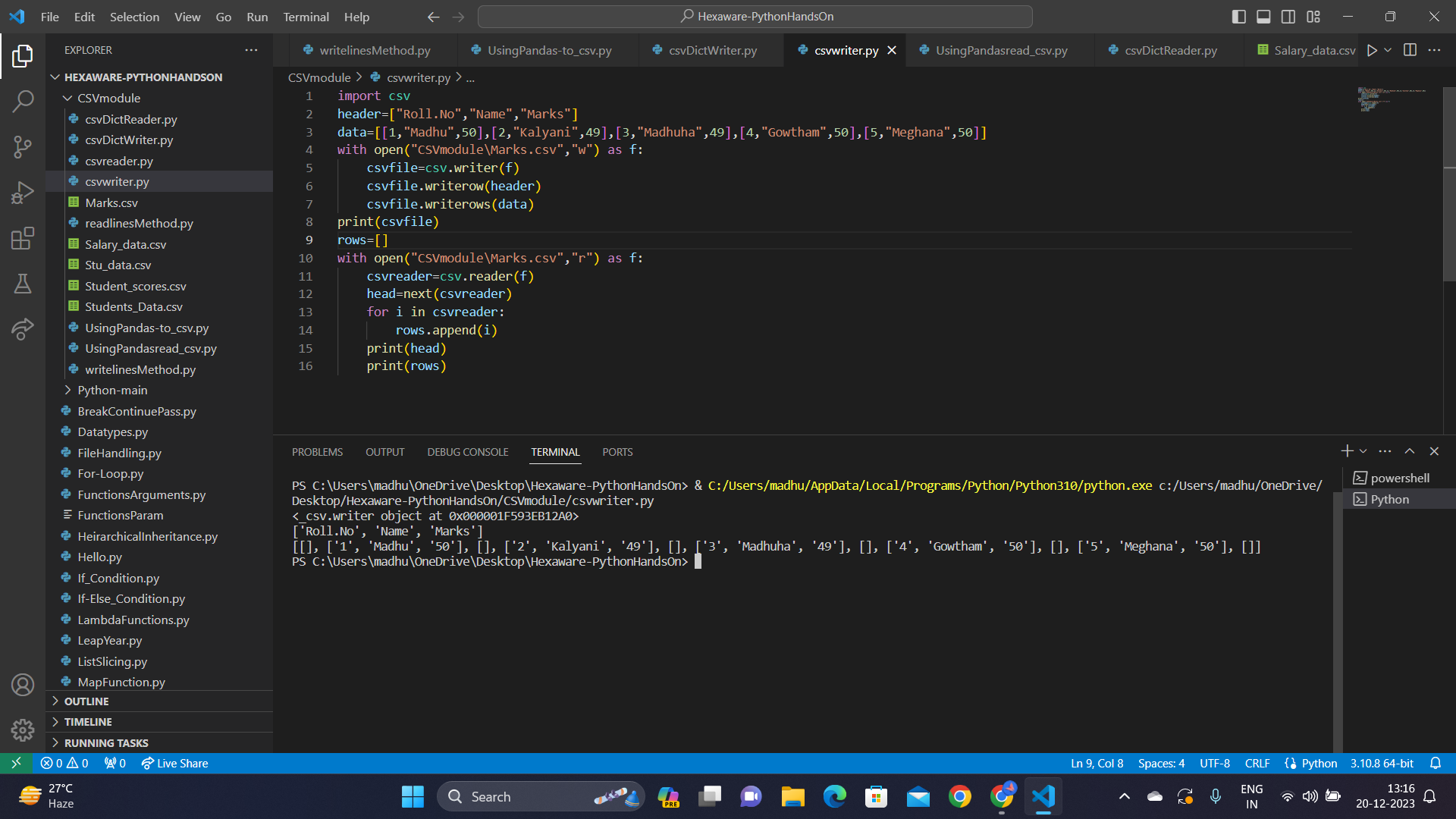
csv.DictReader() is a helper that reads CSV files and organizes the data into labeled dictionaries.



Similarly, There are 4 ways to write csv files in python.

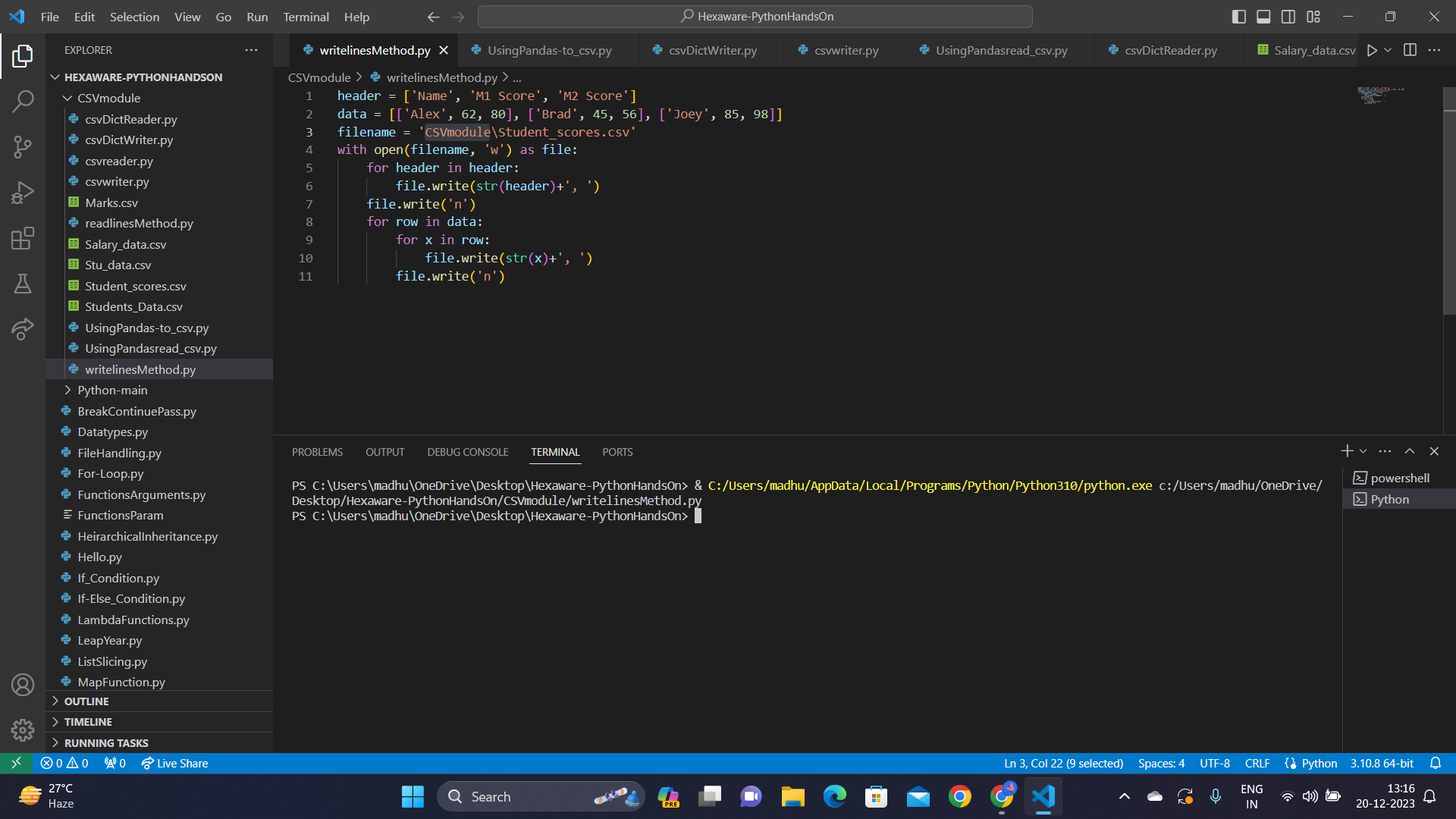
1. Using csv.writer

csv.writer() can write data to a CSV file like creating a list on paper, it helps organize information into rows and columns, making it readable and structured for others to understand easily. It's like making a clear shopping list for your data.



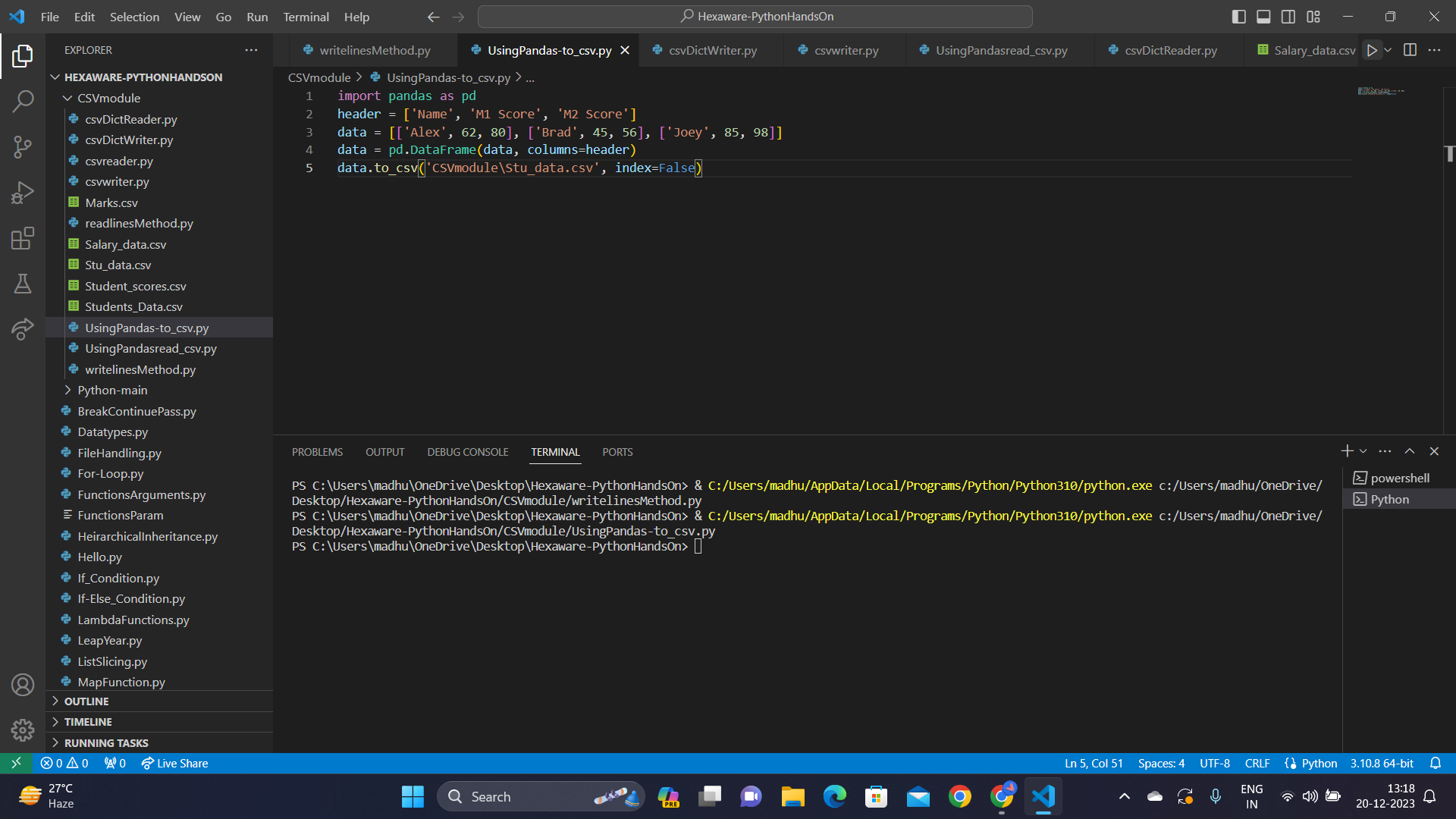
2. Using writelines() function

Using writelines() is like writing a CSV file manually, where you put each line of data one by one, helping you control the format.



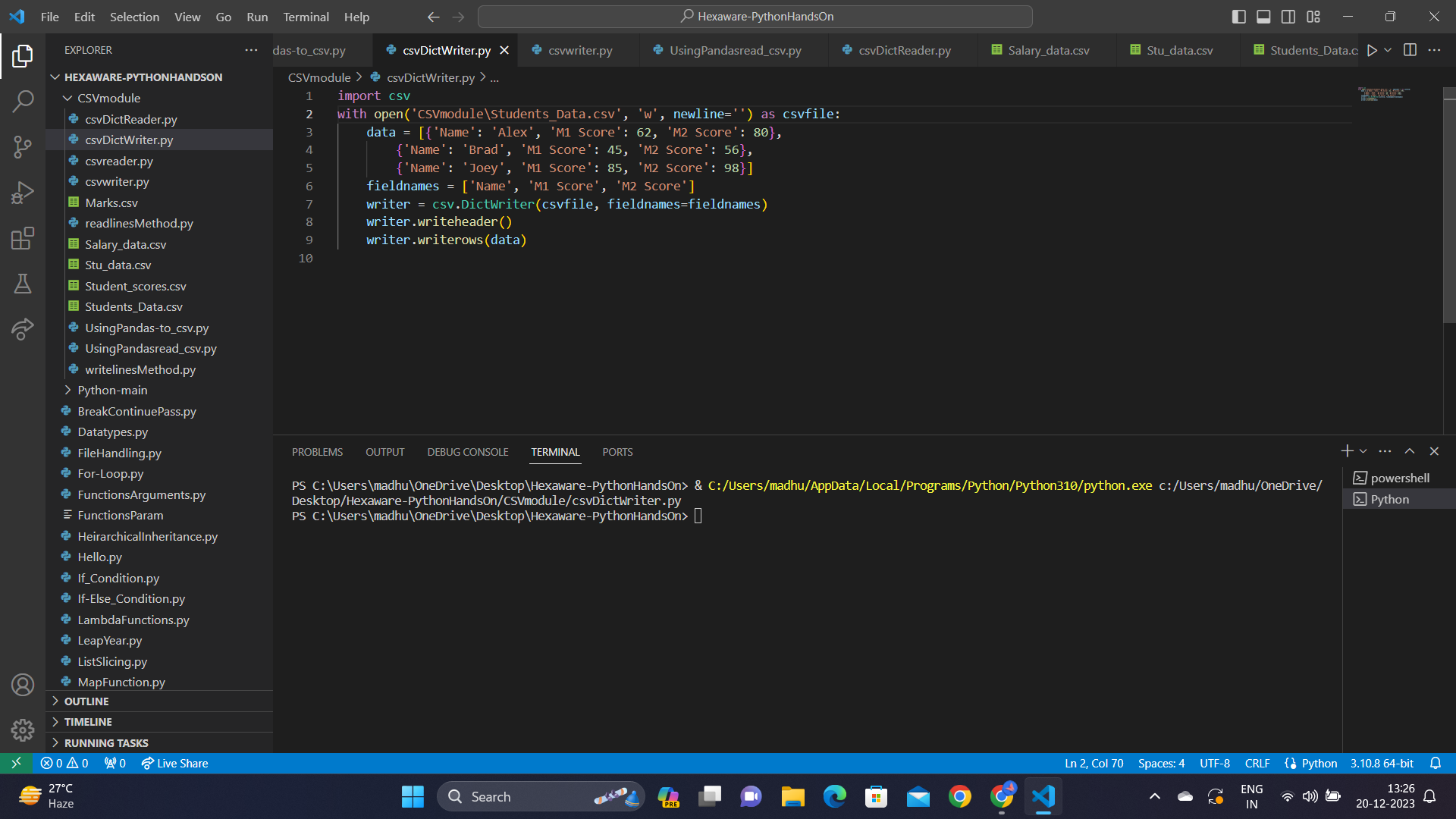
3. Using Pandas

Using Pandas, we create a DataFrame to neatly structure the data and then save it to a CSV file, simplifying the process of sharing and storing information. To achieve this we use to\_csv method.



4. Using csv.DictWriter

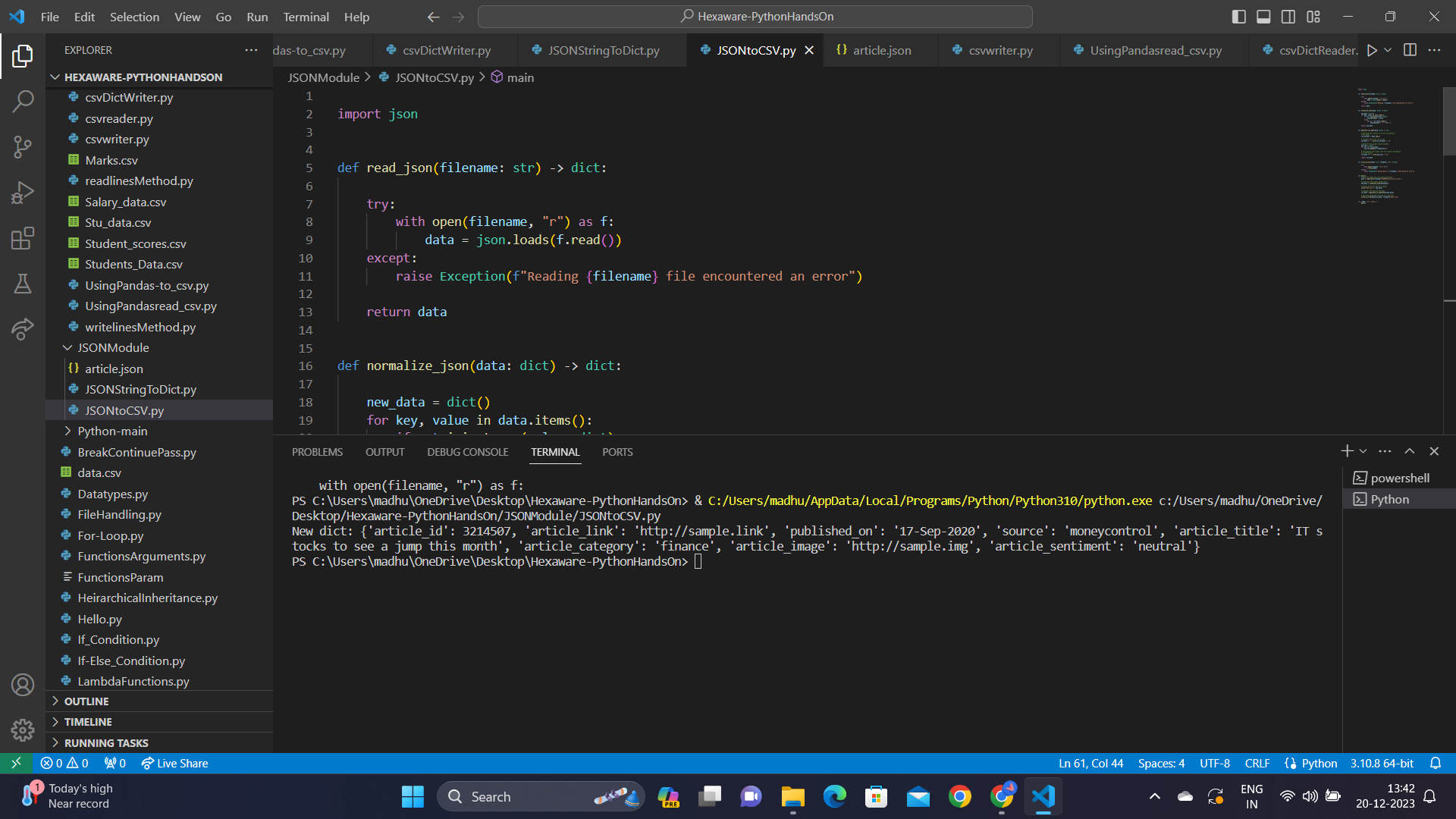
csv.DictWriter() is like creating a personalised blueprint for your data. it organises information in a CSV file with labelled fields, ensuring clarity and making it easy for others to interpret.



JSON

Coverting JSON to CSV:

To convert a JSON file to CSV in Python, we first read the JSON file as a Python dictionary using the read\_json() function. Then, we normalize the dictionary by concatenating nested key strings. The final CSV data is generated by appending records with commas and new lines. Lastly, we write the CSV data to a specified location.



Sorting JSON object in Descending order:

JSON is a data exchange format representing objects and arrays. In Python code, the provided script iterates through student data, printing subject details sorted by code, grade, and enrollment number, demonstrating sorting in descending order.

