100 DAYS 100 PYTHON PROBLEMS

[Day 5: Input and output (IO) operations]

# INPUT OPERATIONS :-

- Reading from the Keyboard: You can use functions or methods to read input from the keyboard in most programming languages. For example, in Python, you can use **input()** to read user input.
- Reading from Files: Reading data from files is a common I/O operation. You can open a file, read its contents, and close it when you're done. Various file modes (e.g., read, write, append) are available.

# OUTPUT OPERATIONS :-

- Writing to the Screen: You can use functions or methods to display information on the screen. For example, in Python, you can use **print()** to display output.
- Writing to Files: Writing data to files is essential for storing information. You can open a file, write to it, and close it. It's important to handle errors and exceptions during file operations.

# HANDLING ERRORS :-

- I/O operations can result in errors, such as file not found, permission denied, or invalid input. You should implement error handling to gracefully manage these situations and provide useful feedback to users.

# DATA FORMATS :-

- Understanding the data formats for I/O is important. For example, when reading data from a file, you should know whether it's in plain text, binary, or a specific format (e.g., JSON, XML, CSV).

100 DAYS 100 PYTHON PROBLEMS

[Day 5: Input and output (IO) operations]

# STREAMS :-

- Data is typically read and written using streams. Streams are sequences of data elements that can be processed sequentially. For example, you can read a file character by character or line by line using streams.

# PRACTICE QUESTIONS :-

1. Write a Python program that reads a text file named "input.txt," counts the number of lines in the file, and then writes the count to an output file named "output.txt."
2. Create a program that asks the user to enter their age. Check if the input is a valid integer. If it's not, handle the exception and ask the user to enter the age again until a valid input is received.
3. Given a JSON file called "data.json" containing an array of objects, write a program that reads the file, extracts specific data (e.g., names or ages), and prints it to the console..
4. Write a Python program that takes multiple text files as input, concatenates them, and saves the result to an output file. The program should ask the user to enter the names of the input files and the name of the output file.
5. Write a program that reads data from a CSV file called "data.csv" and calculates the average of a particular column (e.g., the "scores" column). Print the calculated average to the screen.