

# Assignment Questions File handling

## Open, Read, Write, Append

1. Write a Python script to create a text file named `students.txt` and write 10 student names into it.
  2. Read the contents of `students.txt` and print each line prefixed by the line number.
  3. Append 5 more names to `students.txt` without overwriting existing content.
  4. Write a program that counts the total number of words in a given text file.
  5. Create a log file (`activity.log`) that records the timestamp whenever the script runs.
- 

## File Modes & Pointers

6. Demonstrate the difference between `'r'` and `'r+'` mode by reading and updating a file.
  7. Use `'w+'` mode to create a file, write data, then read it back without closing the file.
  8. Open a file in binary mode (`'rb'`) and display the byte contents.
  9. Create a file, write multiple lines, then use `seek()` to read from a specific position.
  10. Use `tell()` to print the current file pointer position while reading the file.
- 

## File Context Manager (`with` statement)

11. Rewrite Question 1 using a context manager to ensure the file closes automatically.
  12. Create a script that reads a file using a context manager and handles `FileNotFoundError`.
  13. Using a context manager, read `students.txt` and store all lines in a list.
  14. Write a program to safely overwrite a file using a context manager and confirm data is saved.
  15. Demonstrate reading and writing files simultaneously with nested `with` blocks.
- 

## CSV Handling (`csv.reader`, `csv.DictWriter`)

16. Create a CSV file `employees.csv` with headers: Name, Department, Salary.
  17. Write 5 employee records into `employees.csv` using `csv.DictWriter`.
  18. Read `employees.csv` and display all records sorted by Salary descending.
  19. Write a script that filters all employees in the “Sales” department and writes them to a new CSV.
  20. Convert `employees.csv` into a list of dictionaries and print them.
-

## JSON Handling (`json.load`, `json.dumps`)

21. Write a Python dictionary with employee details and save it to `employee.json`.
22. Load `employee.json` and display the data in a formatted way.
23. Update the salary of an employee in `employee.json` and overwrite the file.
24. Write a script to convert `employees.csv` into a JSON file.
25. Validate if a given JSON file is properly formatted or throws `JSONDecodeError`.

---

## Pickle Usage & Security

26. Create a Python dictionary of product inventory and serialize it using `pickle`.
27. Deserialize the pickled data and print it.
28. Modify a record in the deserialized data and overwrite the pickle file.
29. Write a script to load pickle data securely and handle exceptions.
30. Research and explain why loading untrusted pickle files can be a security risk. Demonstrate with an example.

---

## Integration & Realistic Use Cases

31. Create a program that reads names from a text file and writes them to a CSV file.
32. Build a script that appends log entries into a text file whenever a JSON file is updated.
33. Write a program that reads data from a CSV file, transforms it (e.g., increase all salaries by 10%), and writes the results into a new JSON file.
34. Create a backup of any given file by copying its contents to a new file with a timestamp suffix.
35. Develop a script that uses `seek()` to read only the last 100 bytes of a large log file.
36. Create a script that writes user input into a text file until the user types “STOP”.
37. Read a JSON file containing configurations and write specific settings into a `.ini`-style text file.
38. Create a pickle file storing user session data and demonstrate restoring the session state on script restart.
39. Write a script to merge multiple text files into a single file.
40. Create a data export tool that reads product details from JSON, formats them into CSV, and saves them with a timestamped filename.

---

## Sample Data Sets

### Sample Text File: `students.txt`

Alice  
Bob  
Charlie

Diana  
Ethan  
Fiona  
George  
Hannah  
Ian  
Julia

---

### Sample CSV: `employees.csv`

```
Name,Department,Salary
Alice,Sales,55000
Bob,Engineering,75000
Charlie,HR,50000
Diana,Sales,60000
Ethan,Engineering,72000
```

---

### Sample JSON: `employee.json`

```
{
  "employees": [
    {"name": "Alice", "department": "Sales", "salary": 55000},
    {"name": "Bob", "department": "Engineering", "salary": 75000},
    {"name": "Charlie", "department": "HR", "salary": 50000}
  ]
}
```

---

### Sample Pickle Data (Python)

```
import pickle

inventory = {
    "Laptop": {"quantity": 10, "price": 800},
    "Monitor": {"quantity": 20, "price": 150},
    "Mouse": {"quantity": 100, "price": 20}
}

with open("inventory.pkl", "wb") as f:
    pickle.dump(inventory, f)
```