Lab Work – Reading and Writing in Python

Objective

The objective of this lab is to learn and apply file handling in Python to solve practical engineering problems. These tasks help students understand how to read, write, and organize data in CSV and text files, record measurements, and log events.

Task 1 – Load Demand Data Recording

Task: Record hourly load demand of an electrical feeder in a CSV file.

In [1]:
```
import csv

with open("load_demand.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Time", "Load_kW"])
    for hour in range(1, 25):
        writer.writerow([f"{hour}:00", hour * 5])  # Replace with actual load

print("Load demand data recorded! Check load_demand.csv")
```

```
Load demand data recorded! Check load_demand.csv
```

Task 2 – Solar Energy Generation Logging

Task: Record daily solar energy generation and append new values.

In [2]:
```
import csv
from datetime import date

with open("solar_generation.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Date", "Energy_kWh"])
    writer.writerow([date.today(), 50])

# Append new data
with open("solar_generation.csv", "a", newline="") as file:
    writer = csv.writer(file)
    writer.writerow([date.today(), 55])

print("Solar energy data logged! Check solar_generation.csv")
```

```
Solar energy data logged! Check solar_generation.csv
```

Task 3 – Transformer Maintenance Report File

Task: Document transformer maintenance activities in a text file.

In [3]:
```
with open("transformer_maintenance.txt", "w") as file:
```

```
    file.write("Date: 2025-12-20\nActivity: Oil check\nStatus: Completed\n\n")
    file.write("Date: 2025-12-21\nActivity: Insulation test\nStatus: Pending\r

print("Maintenance report created! Check transformer_maintenance.txt")
```

```
Maintenance report created! Check transformer_maintenance.txt
```

Task 4 – Student Energy Audit Data Storage

Task: Store student energy audit records in a CSV file.

In [4]:
```
import csv

with open("student_audit.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Student_Name", "Department", "Audit_Date", "Energy_kWh"]
    writer.writerow(["Ali", "EE", "2025-12-20", 120])
    writer.writerow(["Sara", "EE", "2025-12-20", 110])

print("Student audit data stored! Check student_audit.csv")
```

```
Student audit data stored! Check student_audit.csv
```

Task 5 – Electricity Billing Record Creation

Task: Record monthly electricity billing data and append new data.

In [5]:
```
import csv

with open("billing_data.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Customer", "Month", "Units", "Bill"])
    writer.writerow(["Ali", "Dec", 200, 3000])

with open("billing_data.csv", "a", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Sara", "Dec", 150, 2250])

print("Billing records created! Check billing_data.csv")
```

```
Billing records created! Check billing_data.csv
```

Task 6 – Substation Fault Event Logging

Task: Record fault events and append future events.

In [6]:
```
import csv
from datetime import datetime

with open("fault_log.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Date", "Time", "Fault_Type"])
    writer.writerow([datetime.today().date(), "14:30", "Overload"])
```

```
with open("fault_log.csv", "a", newline="") as file:
    writer = csv.writer(file)
    writer.writerow([datetime.today().date(), "15:00", "Short Circuit"])

print("Fault events logged! Check fault_log.csv")
```

```
Fault events logged! Check fault_log.csv
```