

1. Smart Energy Consumption Analyzer

A smart campus collects **hourly electricity consumption data** (in kWh) for each building.

Task:

Design a Python program that:

- Stores energy consumption using appropriate **variables and data types**
 - Uses a **loop** to process energy data for multiple buildings
 - Uses **if-elif-else** to classify buildings as:
 - *Energy Efficient*
 - *Moderate Consumption*
 - *Energy Intensive*
 - Uses a **function** to compute total and average consumption
 - Uses **break** if total consumption exceeds a critical grid limit
 - Prints a sustainability report
-

2. Carbon Emission Compliance System

An industry must comply with environmental regulations based on **annual CO₂ emissions**.

Task:

Write a Python solution that:

- Accepts emission values for multiple factories
 - Uses **operators** to compute emission growth rate
 - Applies **control flow** to determine penalties or incentives
 - Uses a **loop** to evaluate all factories
 - Skips faulty sensor readings using **continue**
 - Organizes logic using **functions**
 - Clearly explains how the program supports sustainability compliance
-

3. Water Resource Monitoring Program

A region monitors daily water consumption (in liters) from multiple zones.

Task:

Design a Python program that:

- Uses **lists and loops** to store and process daily data

- Uses **functions** to calculate:
 - Daily average consumption
 - Weekly peak demand
 - Uses **if-else** to trigger conservation alerts
 - Uses **pass** for zones where data collection is still under development
 - Outputs recommendations for sustainable water usage
-

4. Renewable Energy Decision Engine (OOP Based)

A smart grid integrates **solar, wind, and hydro** energy sources.

Task:

Using **Object-Oriented Programming**:

- Create a base class **EnergySource**
 - Derive subclasses for Solar, Wind, and Hydro
 - Implement:
 - **Encapsulation** for power output
 - **Inheritance** for shared behavior
 - **Polymorphism** for calculating energy availability
 - Use **conditional logic** to decide which source to prioritize
 - Simulate energy allocation using loops
-

5. Air Quality Index (AQI) Early Warning System

Sensors report AQI values hourly for a city.

Task:

Develop a Python program that:

- Uses **variables and operators** to compute AQI trends
 - Uses **loops** to analyze continuous sensor data
 - Uses **if-elif-else** to classify air quality levels
 - Uses **break** to trigger emergency alerts
 - Uses **functions** to modularize AQI calculations
 - Prints health and sustainability advisories
-

6. Smart Waste Segregation Simulator

A smart waste management system processes waste items as *organic*, *recyclable*, or *hazardous*.

Task:

Create a Python solution that:

- Uses **control flow statements** for waste classification
 - Uses **loops** to process multiple waste entries
 - Uses **functions** to compute recycling efficiency
 - Uses **continue** to skip invalid inputs
 - Uses **OOP** to represent different waste types
 - Justifies how software logic supports environmental sustainability
-

7. Climate Data Integrity Checker

Climate datasets often contain missing or corrupted values.

Task:

Write a Python program that:

- Iterates through temperature and rainfall datasets
 - Uses **conditional checks** to detect anomalies
 - Uses **continue** to ignore corrupted data
 - Uses **functions** to clean and summarize data
 - Uses **operators** to calculate climate trends
 - Explains how such preprocessing is crucial for sustainable policy decisions
-

8. Sustainable Transportation Optimizer

A city wants to promote low-carbon transport.

Task:

Design a Python program that:

- Stores transport data (fuel type, distance, emissions)
- Uses **loops and conditionals** to rank transport modes
- Uses **functions** to compute carbon footprint
- Uses **OOP** to model different vehicle types

- Uses **break** when emissions exceed legal limits
 - Outputs recommendations for sustainable mobility
-

9. Green Data Center Load Manager

A data center wants to reduce energy usage during peak hours.

Task:

Create a Python solution that:

- Monitors server load using **loops**
 - Uses **if-else** to switch to power-saving mode
 - Uses **functions** to calculate energy savings
 - Uses **variables and operators** for threshold checks
 - Explains how software optimization contributes to sustainability
-

10. Open-Ended Design Question

Design a **Python-based sustainability monitoring system** of your choice (energy, water, waste, air, transport).

Your solution must:

- Use **variables, operators, loops, control flow**
- Use **functions and at least one OOP concept**
- Include **break / continue / pass**
- Clearly explain **design decisions and sustainability impact**