|  |  |
| --- | --- |
| Project Title : | Lab Task 3 |
| Name : | **Aman Ali** |
| Roll No : | **195** |
| Class : | BSAI |
| Section : | **3C** |
| Submission Date : | September 3, 2025 |

ModelBaseAgent – Smart AC Controller

**1. Code Explanation**

This program simulates a simple **AI-based AC Controller** that decides whether to turn the AC **ON** or **OFF** based on room temperature.

* It defines a class **ModelBaseAgent** which takes:
  + A **threshold temperature** (default: 23°C)
  + A **file path** to store past temperature decisions.
* Features:  
  • **Retrieve content** → Reads past data (temperature + action) from the file.  
  • **Store content** → Saves the current decision (AC ON or AC OFF) to the file.  
  • **Neurons** → Accepts the current room temperature.  
  • **Operation** → Compares current temperature with threshold and decides:
  + If above threshold → "AC ON KRDO"
  + If equal/below threshold → "AC OFF KRDO"  
    • **Operator** → Checks if this temperature was seen before:
  + If found → Retrieves the past decision from memory.
  + If not → Performs a fresh operation and stores it.
* A dictionary of rooms with their respective temperatures is provided.
* For each room, the program runs the model and prints whether the AC should be **ON** or **OFF**.

**2. How It Works**

* The program starts with a **threshold temperature** (23°C).
* For each room in the dictionary:
  1. A new model object is created.
  2. The room temperature is passed to the **neurons** method.
  3. The **operator** method checks:
     + If the temperature exists in memory (the file).  
       → Prints action from memory.
     + If not found → Calls **operation()**, decides ON/OFF, and stores it in the file.
* Over time, the file acts as the model’s **memory** of past temperature decisions.

**3. Why This Approach**

* Demonstrates the concept of a **basic intelligent agent**.
* Uses **memory storage** to avoid repeating calculations for the same input.
* Simulates **machine learning intuition** → learning from past experiences.
* Provides a real-world use case → a **smart AC system** that automates cooling decisions.
* Uses **OOP concepts** (class, methods, attributes) effectively.

**4. Sample Output**

