**Explanation of the Model-Based Agent Program**

**1. Introduction**

This program demonstrates the working of a **Model-Based Agent** that controls an **Air Conditioner (AC)** based on room temperature.  
The agent takes a fixed threshold temperature as input. For each room, it senses the temperature, decides whether the AC should be turned **ON** or **OFF**, performs the action, and maintains a history of all readings.

**2. Class Definition: Model\_Based\_Agent**

**Constructor: \_\_init\_\_**

def \_\_init\_\_(self, temp, filename="temp\_history.txt"):

self.fixed\_temp = temp

self.filename = filename

open(self.filename, "a").close()

* **self.fixed\_temp**: The reference (threshold) temperature.
* **self.filename**: The file where temperature history will be stored.
* The file is created automatically if it does not already exist.

**Method 1: senser**

def senser(self, temp, room\_name=""):

self.current\_temp = temp

self.current\_room = room\_name

with open(self.filename, "a") as f:

f.write(f"{room\_name}: {temp}\n")

* Captures the **current room temperature** and room name.
* Saves this data into both the agent’s memory and the history file.
* Example file entry:
* Living room: 11
* Drawing room: 24
* Bed Room: 22

**Method 2: performance**

def performance(self):

if self.current\_temp > self.fixed\_temp:

return "Turn ON the AC"

else:

return "Turn OFF the AC"

* Compares the current room temperature with the threshold.
* If the room temperature is **greater** than the threshold → AC should be **ON**.
* Otherwise, AC should be **OFF**.

**Method 3: actuator**

def actuator(self, room\_name=""):

action = self.performance()

print(f"{room\_name} {self.current\_temp} => Action: {action}")

print("History so far:")

with open(self.filename, "r") as f:

history = f.read().strip().split("\n")

print(history)

print("-" \* 40)

* Executes the action decided by the performance method.
* Displays the decision along with the room temperature.
* Prints the complete history of all temperature readings stored in the file.

**3. Main Program**

rooms = {

"Living room": 11,

"Drawing room": 24,

"Bed Room": 22

}

agent = Model\_Based\_Agent(16)

for room, temp in rooms.items():

agent.senser(temp, room)

agent.actuator(room)

agent.senser(22, "Guest Room")

agent.actuator("Guest Room")

* A dictionary of rooms with their respective temperatures is created.
* The agent is initialized with a **threshold temperature of 16°C**.
* For each room:
  + The agent senses the temperature.
  + The agent decides whether to turn the AC **ON** or **OFF**.
  + The decision is displayed along with the updated history.
* Finally, an additional room ("Guest Room") is tested.

**4. Sample Output**

Living room 11 => Action: Turn OFF the AC

History so far:

['Living room: 11']

----------------------------------------

Drawing room 24 => Action: Turn ON the AC

History so far:

['Living room: 11', 'Drawing room: 24']

----------------------------------------

Bed Room 22 => Action: Turn ON the AC

History so far:

['Living room: 11', 'Drawing room: 24', 'Bed Room: 22']

----------------------------------------

Guest Room 22 => Action: Turn ON the AC

History so far:

['Living room: 11', 'Drawing room: 24', 'Bed Room: 22', 'Guest Room: 22']

----------------------------------------