**Name : IMRAN MAJEED**

**Roll NO : 212**

**Section : BSAI-3B**

**Explanation :**

**1. Creating the Agent Class**

class ModelBasedReflexAgent:

def \_\_init\_\_(self, target\_temp):

self.target\_temp = target\_temp

self.current\_temp = None

self.heater\_status = "off"

* This creates a **model-based reflex agent** (a smart temperature controller).
* **target\_temp** → The desired temperature (e.g., 22°C).
* **current\_temp** → The temperature the agent reads from the room.
* **heater\_status** → Keeps track of whether the heater is **on** or **off**.
* When the program starts, the heater is **off**.

### ****2. Sensor Function****

def sensor(self, temp):

self.current\_temp = temp

* The **sensor** receives the **current temperature** of the room.
* Whatever value we give here is stored inside **self.current\_temp** so the agent can make decisions.

### ****3. Decision-Making Function****

def performance(self):

if self.current\_temp < self.target\_temp and self.heater\_status == "off":

self.heater\_status = "on"

return "Turn ON the heater"

elif self.current\_temp >= self.target\_temp and self.heater\_status == "on":

self.heater\_status = "off"

return "Turn OFF the heater"

else:

return "Do nothing (Maintain current state)"

This function **decides what to do** based on two things:

1. **If the room is too cold** (temperature < target) **and** the heater is **off** → **Turn ON the heater**.
2. **If the room is warm enough** (temperature ≥ target) **and** the heater is **on** → **Turn OFF the heater**.
3. **Otherwise** → Do nothing and keep the heater as it is.

This makes the agent **smart** because it **remembers the current state** of the heater.

### ****4. Actuator Function****

def actuator(self):

action = self.performance()

print(f" Current Temp: {self.current\_temp}°C | Heater: {self.heater\_status} | Action: {action}")

* The **actuator** is like the agent’s “hand” — it **performs the action** decided by the performance function.
* It prints:
  + The current temperature
  + The heater’s status (**on/off**)
  + The action taken (**Turn ON, Turn OFF, or Do nothing**)

### ****5. Creating a List of Rooms****

rooms = {

"Living Room": 20,

"Drawing Room": 22,

"Kitchen": 24,

}

* This dictionary contains **room names** and their **current temperatures**.
* For example:
  + **Living Room** → 20°C
  + **Drawing Room** → 22°C
  + **Kitchen** → 24°C

### ****6. Creating the Agent****

agent = ModelBasedReflexAgent(22)

* We create **one agent** that controls all rooms.
* The **target temperature** is set to **22°C**.

### ****7. Testing the Agent in Each Room****

for room, temp in rooms.items():

print(f"{room}:")

agent.sensor(temp)

agent.actuator()

print()

* The program loops through each room.
* The **sensor** reads the room’s temperature.
* The **actuator** decides whether to turn the heater **on**, **off**, or **do nothing**.
* Finally, it prints the result for each room.
* It **decides what to do** using the **performance** function.
* It **controls the heater** using the **actuator**.
* It works for **multiple rooms** but remembers the heater’s state

**OUTPUT :**

