# Model-Based Reflex Agent

## Purpose of Code

The purpose of this code is to implement a Model-Based Reflex Agent that controls a heater based on the current room temperature. The agent not only checks the current temperature but also remembers the previous action it took, so it avoids unnecessary switching of the heater (turning it ON or OFF repeatedly).

## Concepts Used

Artificial Intelligence Agent: An agent that perceives the environment (temperature) and takes an action (turn heater ON/OFF).  
- Model-Based Reflex Agent: Unlike a simple reflex agent, this agent remembers the last action taken and avoids repeating it unnecessarily.  
- Conditional Logic: Used to decide whether to turn the heater ON or OFF.  
- State/Memory: The agent uses a variable (`previous\_action`) to remember past actions.

## Functions & Libraries

- Libraries: This program uses only Python's built-in features (no external libraries).  
- Functions:  
 - \_\_init\_\_: Initializes the agent with a fixed temperature and memory.  
 - sensor: Reads the current temperature from the environment.  
 - performance: Decides whether to turn the heater ON, OFF, or do nothing, based on the current temperature and previous action.  
 - actuator: Executes the action and prints the result.

## Python Code

class ModelBasedReflexAgent:  
 def \_\_init\_\_(self, temp):  
 self.fixed\_temp = temp  
 self.previous\_action = None  
  
 def sensor(self, temp):  
 self.current\_temp = temp  
  
 def performance(self):  
 # Step 1: Decide action  
 if self.current\_temp > self.fixed\_temp:  
 action = "Turn off the Heater"  
 else:  
 action = "Turn on the Heater"  
  
 # Step 2: Avoid unnecessary switching (applies to both cases)  
 if action == self.previous\_action:  
 action = "No action needed"  
 else:  
 self.previous\_action = action  
  
 return action  
  
 def actuator(self):  
 action = self.performance()  
 print(self.current\_temp, "=> Action:", action)  
  
agent = ModelBasedReflexAgent(18)  
rooms = {  
 "Living room": 30,  
 "Drawing room": 24,  
 "Bed room": 16,  
 "Kitchen": 34,  
}  
  
for room, temp in rooms.items():  
 print(room, end=":\t")  
 agent.sensor(temp)  
 agent.actuator()