***Name : Ahmad Raza***

***Roll Nob: 093***

**Line-by-Line Explanation of the Code**

**Line 1:**

**class AC\_Agent:**

**This line defines a new class called AC\_Agent, which represents an intelligent agent that controls the air conditioner based on a set temperature.**

**Line 2:**

**def \_\_init\_\_ (self, set\_temp):**

**This is the constructor method for the AC\_Agent class. It is automatically called when an object of this class is created. It takes one parameter set\_temp which is the desired temperature to maintain.**

**Line 3:**

**self.set\_temp = set\_temp**

**The set\_temp value passed during object creation is stored in the instance variable self.set\_temp for future reference.**

**Line 4:**

**self.last\_move = None**

**This initializes another instance variable last\_move, which keeps track of the last action taken by the agent (whether the AC was turned ON or OFF). It is set to None initially because no action has been taken yet.**

**Line 5:**

**def check(self, temp):**

**This defines a method named check inside the AC\_Agent class. It takes the current temperature temp as input and decides whether to turn the AC ON or OFF.**

**Line 6-7:**

**if temp > self.set\_temp:**

**move = "AC is ON"**

**If the current temperature is greater than the set temperature, the AC should be turned ON. So, the variable move is set to "AC is ON".**

**Line 8-9:**

**else:**

**move = "AC is OFF"**

**If the current temperature is not greater than the set temperature (i.e., it's equal to or less than), the AC should be turned OFF. The move variable is set to "AC is OFF".**

**Line 10-11:**

**if move == self.last\_move:**

**move = print(f"No change, keep {self.last\_move}")**

**If the current decision (move) is the same as the previous decision (self.last\_move), it prints a message saying that there's no change and retains the previous action. It also replaces the value of move with the result of print(), which is None.**

**Line 12-13:**

**if "No change" not in move:**

**self.last\_move = move**

**If the word "No change" is not in move, meaning the action has changed, then it updates self.last\_move with the new action. Otherwise, it keeps the previous state unchanged.**

**Line 16-20:**

**areas = {**

**"Study Room": 27,**

**"Meeting Room": 22,**

**"Dining Hall": 25,**

**"Garage": 20**

**}**

**This dictionary named areas defines different rooms and their current temperatures in °C.**

**Line 22:**

**set\_temp = 24**

**This line sets the desired temperature (set\_temp) to 24°C. The AC agent will use this as the target for comparison.**

**Line 23:**

**bot = AC\_Agent(set\_temp)**

**Here, an instance of AC\_Agent named bot is created using the set temperature defined earlier.**

**Line 26-27:**

**for place, temp in areas.items():**

**result = bot.check(temp)**

**A loop is used to go through each room in the areas dictionary. For each room (place) and its temperature (temp), the check() method is called.**

**Line 28:**

**print(f"{place}: {temp}°C  is {result}")**

**This prints the result for each room, showing the room name, current temperature, and the result returned from the check() method.  
However, check() does not return anything, so result will always be None. The actual message is printed inside the check() method using print().**