

**Name : Izaan Mohtashim**

**Roll No: P20-0613**

**Sec : 6-C**

**Sub: Artificial Intelligent LAB**

# LAB TASK 2



## TASK-1

```
In [7]: class fire_alarm:
        def __init__(self):
            self.state = "normal"

        def set_alarm(self,temp,smoke):
            if smoke:
                if temp < 50:
                    print("Smoke is there with normal temperature ")
                    self.state = "Smokyyyyyy "
                    self.sound_alarm()
                    self.sprinkle_sys()
                else:
                    print("Smoke and temperature is greater than 50")
                    self.state = "smoke and High Temp "
                    self.sound_alarm()
                    self.call_fire_dept()
            else:
                if temp < 50:
                    print("smoke not detected with normal temperature ")
                    self.state = "Normal temperature with no smoke"
                else:
                    print("No Smoke and temperature is greater than 50")
                    self.state = "No smoke and High Temp "
                    self.sound_alarm()
                    self.call_fire_dept()
                    self.sprinkle_sys()
```

```
def sound_alarm(self):
    print("ALARAM SOUND ON.....\n")
def sprinkle_sys(self):
    print("SPRINLE SYSTEM ACTIVATED*****")
def call_fire_dept(self):
    print("CALLING THE FIRE DEPT*****")
def __str__(self):
    return f" state {self.state}"
```

```
fire_alarm = fire_alarm()

temperature = int(input("ENTER THE TEMPERATURE"))
Smoke = input("Is there smoke in room (Yes or No):")

fire_alarm.set_alarm(temperature,Smoke)
print(fire_alarm)
```

```
ENTER THE TEMPERATURE45
Is there smoke in room (Yes or No):YES
Smoke is there with normal temperature
ALARAM SOUND ON.....
```

```
SPRINLE SYSTEM ACTIVATED*****
state Smokyyyyyy
```

## TASK-2

```
In [8]: import logging
class Moisture_Sensor:

    #setting the initial state
    def __init__(self, threshold_dry, threshold_moist, threshold_wet):
        self.threshold_dry = threshold_dry
        self.threshold_moist = threshold_moist
        self.threshold_wet = threshold_wet

    #Comparing to check the current state
    def read_moisture(self):
        try:
            # Read moisture level from the sensor
            moisture_level = float(input("Enter the moisture level: "))
            if moisture_level <= self.threshold_dry:
                return "Dry soil"
            elif moisture_level <= self.threshold_moist:
                return "Moist soil"
            elif moisture_level <= self.threshold_wet:
                return "Wet soil"
            else:
                raise ValueError("Invalid moisture level")
        except Exception as e:
            logging.error("Error reading moisture level: %s", e)

class Water_Control_Unit:
    def __init__(self, sensor):
        self.sensor = sensor

    def run(self):
        moisture_state = self.sensor.read_moisture()
        if moisture_state == "Dry soil":
            self.water_plants()
        elif moisture_state == "Moist soil":
            self.do_nothing()
```

```
        elif moisture_state == "Wet soil":
            self.stop_watering()
        else:
            raise ValueError("Invalid moisture state")

    #Function for printing
    def water_plants(self):
        print("Watering plants...")

    def do_nothing(self):
        print("Moisture level is adequate, doing nothing...")

    def stop_watering(self):
        print("Moisture level is high, stopping watering...")

#Main function and setting threshold value
if __name__ == "__main__":
    logging.basicConfig(filename="watering_system.log", level=logging.ERROR)

    # Set threshold values for each soil state
    threshold_dry = 20.0
    threshold_moist = 40.0
    threshold_wet = 60.0

    sensor = Moisture_Sensor(threshold_dry, threshold_moist, threshold_wet)
    control_unit = Water_Control_Unit(sensor)
    control_unit.run()
```

Enter the moisture level: 34  
Moisture level is adequate, doing nothing...