

SOURCE CODE :-

```
import time

# Sensor threshold values
MOISTURE_THRESHOLD = 30 # Threshold for soil moisture (in percentage)
TEMPERATURE_THRESHOLD = 35 # Threshold for temperature (in Celsius)

# Mock sensor classes
class SoilMoistureSensor:
    def __init__(self):
        self.moisture_level = 50 # Initial moisture level

    def read(self):
        # Simulate reading from a soil moisture sensor
        return self.moisture_level

    def decrease_moisture(self):
        # Simulate moisture reduction over time
        self.moisture_level -= 5

class TemperatureSensor:
    def __init__(self):
        self.temperature = 25 # Initial temperature

    def read(self):
        # Simulate reading from a temperature sensor
        return self.temperature

    def increase_temperature(self):
        # Simulate temperature increase over time
        self.temperature += 1

# Mock actuator class
class WaterPump:
    def __init__(self):
        self.is_on = False

    def turn_on(self):
        self.is_on = True
        print("Water pump turned ON")
```

```
def turn_off(self):
    self.is_on = False
    print("Water pump turned OFF")
```

```
# Main Smart Irrigation System class
```

```
class SmartIrrigationSystem:
```

```
    def __init__(self, moisture_sensor, temperature_sensor, water_pump):
        self.moisture_sensor = moisture_sensor
        self.temperature_sensor = temperature_sensor
        self.water_pump = water_pump
```

```
    def control_irrigation(self):
        moisture_level = self.moisture_sensor.read()
        temperature = self.temperature_sensor.read()
```

```
        print(f"Soil moisture level: {moisture_level}%")
        print(f"Temperature: {temperature}°C")
```

```
        if moisture_level < MOISTURE_THRESHOLD and temperature <
TEMPERATURE_THRESHOLD:
```

```
            self.water_pump.turn_on()
        else:
            self.water_pump.turn_off()
```

```
# Main function to simulate the irrigation process
```

```
def main():
```

```
    # Create sensor and actuator objects
```

```
    moisture_sensor = SoilMoistureSensor()
```

```
    temperature_sensor = TemperatureSensor()
```

```
    water_pump = WaterPump()
```

```
    # Create the smart irrigation system
```

```
    irrigation_system = SmartIrrigationSystem(moisture_sensor, temperature_sensor,
water_pump)
```

```
    # Simulate the system running over time
```

```
    for _ in range(10): # Simulate 10 cycles
```

```
        irrigation_system.control_irrigation()
```

```
        moisture_sensor.decrease_moisture() # Simulate soil drying out
```

```
        temperature_sensor.increase_temperature() # Simulate temperature increasing
        time.sleep(5) # Wait for 5 seconds before the next cycle
```

```
if __name__ == "__main__":  
    main()
```