**import java.util.Scanner;**

**public class AutomaticWateringSystem {**

**// Define threshold for moisture level (range from 0 to 1023 for most soil moisture sensors)**

**static final int MOISTURE\_THRESHOLD = 500; // You can adjust this value depending on your test setup**

**// Simulated soil moisture sensor input (0 to 1023 range)**

**static int soilMoistureLevel = 0; // Default moisture level (simulate it)**

**// Simulated water pump status**

**static boolean waterPumpStatus = false;**

**public static void main(String[] args) {**

**// Scanner to simulate sensor reading**

**Scanner scanner = new Scanner(System.in);**

**// Main loop for controlling watering system**

**while (true) {**

**// Simulate the process of reading moisture level from sensor (input from user for now)**

**System.out.println("Enter the soil moisture level (0 to 1023): ");**

**soilMoistureLevel = scanner.nextInt();**

**// Check the moisture level and control the water pump accordingly**

**if (soilMoistureLevel < MOISTURE\_THRESHOLD) {**

**// Soil is dry, turn on water pump**

**waterPumpStatus = true;**

**System.out.println("Soil moisture is low. Turning on the water pump...");**

**waterPlants();**

**} else {**

**// Soil is wet enough, turn off water pump**

**waterPumpStatus = false;**

**System.out.println("Soil moisture is sufficient. Turning off the water pump...");**

**stopWatering();**

**}**

**// Simulate a delay between sensor readings (e.g., 5 seconds)**

**try {**

**Thread.sleep(5000); // Wait for 5 seconds before the next reading**

**} catch (InterruptedException e) {**

**System.out.println("Error: " + e.getMessage());**

**}**

**}**

**}**

**// Function to simulate watering the plants**

**private static void waterPlants() {**

**if (waterPumpStatus) {**

**System.out.println("Water pump is ON. Watering the plants...");**

**// Simulate the watering process (can be extended to set time for watering)**

**try {**

**Thread.sleep(2000); // Simulate 2 seconds of watering**

**System.out.println("Watering complete.");**

**} catch (InterruptedException e) {**

**System.out.println("Error while watering: " + e.getMessage());**

**}**

**}**

**}**

**// Function to stop watering the plants**

**private static void stopWatering() {**

**if (!waterPumpStatus) {**

**System.out.println("Water pump is OFF. No watering needed.");**

**}**

**}**

**}**