

1<sup>st</sup> year master Internet of Things & Cybersecurity**Solution :****Exercise 1 :**

```
// Include necessary libraries
#include <DHT.h>
// Define sensor pin and type
#define DHT_PIN 2
#define DHT_TYPE DHT22
// Define LED pin
#define LED_PIN 13
// Initialize DHT sensor
DHT dht(DHT_PIN, DHT_TYPE);
void setup() {
  // Start serial communication
  Serial.begin(9600);
  // Initialize DHT sensor
  dht.begin();
  // Set LED pin as output
  pinMode(LED_PIN, OUTPUT);
}
void loop() {
  // Read sensor values
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  // Display data on serial monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C, Humidity: ");
  Serial.print(humidity);
  Serial.println("%");
  // Check if temperature exceeds a threshold
  if (temperature > 25.0) {
    digitalWrite(LED_PIN, HIGH); // Turn on LED
    Serial.println("Warning: High Temperature!");
  } else {
    digitalWrite(LED_PIN, LOW); // Turn off LED
  }
  // Delay for better readability (adjust as needed)
  delay(2000);
}
```

**Exercise 2 :**

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include <ESP8266WiFi.h>
#include <ESP8266SMTP.h>
#define PIR_PIN D2
#define DOOR_SENSOR_PIN D3
#define LED_PIN 13
const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";
const char* smtpServer = "YourSMTPServer";
const char* emailSender = "YourEmail@gmail.com";
```

1<sup>st</sup> year master Internet of Things & Cybersecurity

```

const char* emailRecipient = "RecipientEmail@gmail.com";
const char* emailSubject = "Security Alert";
const char* emailBody = "Motion detected or door/window opened. Attached is the captured image.";

Adafruit_MLX90614 mlx = Adafruit_MLX90614();
void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(DOOR_SENSOR_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}
void loop() {
  if (motionDetected() || doorWindowOpened()) {
    captureAndSendAlert();
  }
}
bool motionDetected() {
  return digitalRead(PIR_PIN) == HIGH;
}
bool doorWindowOpened() {
  return digitalRead(DOOR_SENSOR_PIN) == HIGH;
}
void captureAndSendAlert() {
  digitalWrite(LED_PIN, HIGH);
  // Capture image using camera module (implementation depends on the camera library used)
  // Send email alert
  ESP8266SMTP::QuickSend("your.smtp.server", 25, "your.email@gmail.com", "recipient.email@example.com",
"Subject", "Body");
  digitalWrite(LED_PIN, LOW);
  delay(5000); // Delay to avoid multiple alerts for the same event
}

```

**Exercise 3 :**

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define SOIL_MOISTURE_PIN A0
#define LIGHT_SENSOR_PIN A1
#define TEMPERATURE_SENSOR_PIN A2
#define WATER_LED_PIN 7
#define LIGHT_LED_PIN 8
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(WATER_LED_PIN, OUTPUT);
  pinMode(LIGHT_LED_PIN, OUTPUT);
}

```

1<sup>st</sup> year master Internet of Things & Cybersecurity

---

```
}  
void loop() {  
  // Read sensor values  
  int soilMoisture = analogRead(SOIL_MOISTURE_PIN);  
  int lightValue = analogRead(LIGHT_SENSOR_PIN);  
  float temperature = analogRead(TEMPERATURE_SENSOR_PIN) * 0.488; // Conversion for LM35 sensor  
  // Display data on LCD  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("Moisture: ");  
  lcd.print(soilMoisture);  
  lcd.setCursor(0, 1);  
  lcd.print("Light: ");  
  lcd.print(lightValue);  
  lcd.print(" Temp: ");  
  lcd.print(temperature);  
  // Check conditions and display alerts  
  if (soilMoisture < 300) {  
    digitalWrite(WATER_LED_PIN, HIGH); // Turn on watering LED  
    Serial.println("Plant needs watering!");  
  } else {  
    digitalWrite(WATER_LED_PIN, LOW); // Turn off watering LED  
  }  
  if (lightValue < 500) {  
    digitalWrite(LIGHT_LED_PIN, HIGH); // Turn on light condition LED  
    Serial.println("Low light conditions!");  
  } else {  
    digitalWrite(LIGHT_LED_PIN, LOW); // Turn off light condition LED  
  }  
  delay(2000); // Adjust delay based on your preference  
}
```