```cpp
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Pin for the LM35 temperature sensor
const int tempPin = A0;

// Variables for temperature readings
float temperatureC = 0.0;
float voltage = 0.0;
int sensorValue = 0;

// Create an LCD object with I2C address 0x27 (check your LCD address if different)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Define temperature thresholds for alerts (e.g., over 30°C or below 10°C)
const float upperThreshold = 30.0;
const float lowerThreshold = 10.0;

// Define status flags for alerts
bool temperatureWarning = false;

void setup() {
  // Start serial communication for debugging
  Serial.begin(9600);

  // Initialize the LCD screen
  lcd.begin();
  lcd.backlight(); // Turn on the backlight

  // Display a welcome message on the LCD
  displayCustomMessage("Temp Monitoring\nSystem Initializing");
  delay(2000); // Display the message for 2 seconds
  lcd.clear();

  // Initialize sensor readings
  initializeSensor();
}

void loop() {
  // Read temperature sensor data
  sensorValue = analogRead(tempPin);
  voltage = sensorValue * (5.0 / 1023.0); // Convert sensor value to voltage
  temperatureC = voltage * 100.0; // Convert voltage to temperature in Celsius (LM35)
```

```
  // Display temperature on Serial Monitor for debugging
  Serial.print("Sensor Value: ");
  Serial.print(sensorValue);
  Serial.print(" - Voltage: ");
  Serial.print(voltage);
  Serial.print("V - Temperature: ");
  Serial.print(temperatureC);
  Serial.println(" C");

  // Check if the temperature exceeds defined limits
  checkTemperatureAlert(temperatureC);

  // Display the temperature on the LCD
  updateLCD("Temp: " + String(temperatureC) + " C", 0);

  // Show alert messages if temperature is too high or low
  if (temperatureWarning) {
    updateLCD("ALERT! Temp Issue", 1);
  } else {
    updateLCD("System Normal", 1);
  }

  // Wait 1 second before taking the next reading
  delay(1000);
}

// Function to check if the temperature exceeds the threshold
void checkTemperatureAlert(float currentTemp) {
  // If the temperature is higher than the upper threshold
  if (currentTemp > upperThreshold) {
    temperatureWarning = true;
    Serial.println("Warning: Temperature is too high!");
  }
  // If the temperature is lower than the lower threshold
  else if (currentTemp < lowerThreshold) {
    temperatureWarning = true;
    Serial.println("Warning: Temperature is too low!");
  }
  // If the temperature is within the normal range
  else {
    temperatureWarning = false;
  }
}
```

```cpp
// Function to initialize the sensor with a brief calibration
void initializeSensor() {
  Serial.println("Initializing Temperature Sensor...");
  updateLCD("Sensor Initializing", 1);
  delay(1000); // Delay to simulate sensor initialization time
  Serial.println("Sensor Initialized.");
  lcd.clear();
}

// Function to handle LCD updates efficiently
void updateLCD(String message, int line) {
  // Clear the specified line on the LCD before updating
  lcd.setCursor(0, line);
  lcd.print("          "); // Clear the line with spaces
  lcd.setCursor(0, line);
  lcd.print(message); // Print the new message on the specified line
}

// Function to display a custom message on the LCD
void displayCustomMessage(String message) {
  lcd.clear();
  int line = 0;
  int pos = 0;
  while (pos < message.length()) {
    int nextPos = message.indexOf('\n', pos);
    if (nextPos == -1) nextPos = message.length();
    lcd.setCursor(0, line);
    lcd.print(message.substring(pos, nextPos));
    pos = nextPos + 1;
    line++;
  }
  delay(2000); // Display custom message for 2 seconds
}
```

Output
Sketch uses 9906 bytes (30%) of program storage space. Maximum is 32256 bytes.
Global variables use 640 bytes (31%) of dynamic memory, leaving 1408 bytes for local variables. Maximum is 2048 bytes.