[CS-350-11992-M01 Emerging Sys Arch & Tech 2025 C](https://learn.snhu.edu/d2l/home/1918341)

CS 350

Module 7

7-1 Final Project: Thermostat Lab

CS 350 Final Project

Title: CS 350 Final Project – Thermostat Prototype  
Author: Marissa Lanza  
Date: June 23, 2025

**Introduction**

This prototype implements the low-level logic of a three-state (OFF/HEATING/COOLING) thermostat entirely in Python. Because I did **not** have a Raspberry Pi or the AHT20 sensor and LED/button hardware on hand, I developed *import stubs* for all GPIO, I²C, and UART libraries. I confirmed this approach with my professor, who indicated that the primary deliverables are:

1. **The Python code** (main.py),
2. **The state-machine diagram** exported as PDF (state\_machine.pdf), and
3. **This written reflection** (.docx).

**Functionality & Stub Strategy**

* **State Machine** cycles through OFF → HEATING → COOLING → OFF via a “mode” button stub.
* **Set-Point Adjustment** with “Up”/“Down” button stubs.
* **Temperature Reading** returns a placeholder drift (72–73 °F); on actual hardware, replace with real AHT20 I²C code.
* **LED Output** simulated by no-op PWMLED stubs. Heating pulses red, cooling pulses blue, solid when at set-point.
* **Display** uses print() to simulate the LCD (timestamp, mode, temperatures).
* **UART** stub writes comma-delimited status strings to a dummy serial port, simulating server communication.

All hardware-dependent imports are wrapped in try/except ImportError so the logic can be fully tested on Windows.

**State-Machine Diagram**

*(Include the one-page PDF you exported from draw.io here.)*  
The diagram shows transitions on mode‐button presses and how each state maps to the LED and temperature-comparison logic.

**Code Quality & Best Practices**

* **Modular design**: separate methods for sensing, actuation, display, and communication.
* **Descriptive naming**: e.g. toggle\_mode(), update\_leds().
* **In-line comments** clarify sensor stubs and timing choices.
* **Testability**: shortened loop delay (time.sleep(5)) for rapid local testing.

**Next Steps & Hardware Integration**

1. **When hardware arrives**: remove stubs, install gpiozero, smbus2, and pyserial, hook up buttons/LEDs/AHT20 per lab guide.
2. **LCD driver**: swap print() for an actual LCD‐driver library.
3. **Cloud connectivity**: extend the UART stub to real Wi-Fi (e.g. MQTT).
4. **Unit tests**: add pytest or unittest coverage for state transitions.

My PDF of thermostat lab:

A diagram of a heating system

AI-generated content may be incorrect.