

# Simulating a Temperature Sensor with User Control

## Introduction

In this program, we have developed a simple temperature control system that simulates a temperature sensor. The goal of this system is to generate random temperature values within a predefined range while allowing user interaction to modify the temperature or exit the simulation. The program uses functions for simulation, user interaction, and control flow, creating temperature monitoring experience.

## Program Structure

The program consists of three primary functions:

1. `fakenews()`: Simulates the temperature fluctuations.
2. `view()`: Displays the initial message and the current temperature.
3. `den()`: Manages the interaction between the temperature sensor simulation and the user inputs.

## Function Descriptions

### 1. `fakenews()`

The `fakenews()` function acts as a **model** for this program. It simulates the behavior of a sensor providing random values in a specified range (55°F to 85°F), which is typical for a home environment. The function works as follows:

- The current temperature starts at 72°F, a standard room temperature.
- The function generates a random number between 0 and 9 using `rand() % 10`.
- If the random number is 0, the temperature decreases by 1°F. If it is 9, the temperature increases by 1°F. This randomness simulates real-world fluctuations in temperature due to environmental factors.
- The temperature is checked against the minimum (55°F) and maximum (85°F) bounds to ensure it stays within the specified range.
- The updated temperature is returned and printed for the user.

### 2. `view()`

The `view()` function provides the user with an introduction to the system. It prints the initial temperature setting and offers instructions on how to interact with the program:

- The current temperature is displayed.
- The user is informed that they can press 't' to set a new temperature or 'x' to exit the program.

This function is primarily used to give the user a clear understanding of how to interact with the temperature control system.

### 3. den()

The den() function manages the main logic of the program. It continuously monitors the simulated temperature, displays the current temperature, and registers for user input.

The function operates in an infinite loop. It performs the following steps:

1. **Call to fakenews():** The program repeatedly calls fakenews() to update the temperature and display the new value. This simulates an ongoing temperature monitoring system.
2. **User Input Handling:** To enable user interaction, the program waits for key presses. In Windows systems, the \_kbhit() function is used to detect if a key has been pressed.
3. **Responding to Input:**
  - If the user presses 't', the program prompts for a new temperature. The input is then validated to ensure it remains within the allowable range.
  - If the user presses 'x', the program terminates using the exit(0) function, ending the simulation gracefully.
4. **Delays:** After each iteration, the program pauses for one second using the sleep() function.

### 4. Conclusion

The program achieves the following objectives:

- Simulating a fluctuating temperature sensor that generates random values between 55°F and 85°F.
- Allowing user interaction to manually set the temperature or exit the simulation.