Exercise 2 (Windows): UDP Sensor Client-Server

Objective

Learn how to use **UDP sockets** in C++ on Windows (Winsock2) to send and receive simple sensor data.

Task

You will write two C++ programs:

1. Sensor Client

- Sends a UDP packet every second.
- The packet contains a simple message like: `Temperature: 23°C`.

2. Server

- Listens on a UDP port.
- Prints every message received.

Optional Challenge

- Add a simple checksum to the message.
- Make the client resend the message if no acknowledgment is received within 1 second.

Starter Code

sensor_client.cpp (Stub)

```
#include <iostream>
#include <winsock2.h>
#include <ws2tcpip.h>

#pragma comment(lib, "ws2_32.lib")

int main() {
    // TODO: Initialize Winsock

    // TODO: Create UDP socket

    // TODO: Fill in server information (IP, Port)
```

```
    while (true) {
        // TODO: Create message

        // TODO: Send message to server

        // TODO: Sleep for 1 second
    }

    // TODO: Cleanup Winsock

    return 0;
}
```

sensor_server.cpp (Stub)

```
#include <iostream>
#include <winsock2.h>
#include <ws2tcpip.h>

#pragma comment(lib, "ws2_32.lib")

int main() {
    // TODO: Initialize Winsock

    // TODO: Create UDP socket

    // TODO: Bind to a port

    while (true) {
        char buffer[1024];
        struct sockaddr_in client_addr;
        int addr_len = sizeof(client_addr);

        // TODO: Receive message from client

        // TODO: Print received message

        // TODO (Optional): Send acknowledgment
    }

    // TODO: Cleanup Winsock

    return 0;
}
```

Learning Goals

- Understand how UDP works at the socket level.
- Practice sending and receiving data over the network using Winsock.
- Realize the stateless and unreliable nature of UDP.

Suggested Questions for Students

1. What is the difference between UDP and TCP communication?
2. How does the server receive the client's IP address?
3. What happens if the UDP packet is lost?

Hint

You will need to look up these functions:

- `WSAStartup()` / `WSACleanup()`
- `socket()`
- `sendto()`
- `recvfrom()`
- `bind()`
- `closesocket()