

Exercise 3 (Windows) : TCP Client-Server Echo Program

Objective

Learn how to use **TCP sockets** in C++ on Windows (Winsock2) to establish a connection and exchange data reliably.

Task

You will write two C++ programs:

1. Echo Server

- Listens on a TCP port (e.g., 5001).
- Accepts client connections.
- Receives a message and sends the **same message back** (echo).

2. Echo Client

- Connects to the server.
- Sends a message (e.g., "Hello Server!").
- Receives the echoed message and prints it.

Optional Challenge

- Make the server handle multiple clients concurrently (using threads).
- Add simple error handling for disconnected clients.

Starter Code

echo_server.cpp (Stub)

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

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

int main() {
    // TODO: Initialize Winsock

    // TODO: Create TCP socket
```

```

    // TODO: Bind to a port

    // TODO: Listen for connections

    // TODO: Accept client connection

    // TODO: Receive message from client

    // TODO: Send the same message back (echo)

    // TODO: Cleanup Winsock

    return 0;
}

```

echo_client.cpp (Stub)

```

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

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

int main() {
    // TODO: Initialize Winsock

    // TODO: Create TCP socket

    // TODO: Connect to server

    // TODO: Send message to server

    // TODO: Receive echo message

    // TODO: Cleanup Winsock

    return 0;
}

```

Learning Goals

- Understand the difference between **UDP** and **TCP** communication.

- Practice establishing and maintaining a **connection-oriented** communication channel.
- Learn how to use the Winsock2 API for TCP sockets.

Suggested Questions for Students

1. What steps are required to establish a TCP connection on Windows?
2. How does TCP differ from UDP?
3. How can you make the server handle multiple clients?

Hint

You will need to look up these functions:

- `WSAStartup()` / `WSACleanup()`
- `socket()`
- `bind()`
- `listen()`
- `accept()`
- `connect()`
- `send()`
- `recv()`
- `closesocket()`