

Echo Server (Client)

Description

In this programming challenge, you are tasked with implementing a basic "echo server". An echo server simply sends back the same message it receives from a client. This requires setting up both a server and a client. The client sends a message to the server, the server receives this message, and then the server sends the same message back to the client. The client then verifies the received message matches the one sent.

Input

No direct input is provided through standard input as this challenge involves network communication. The client will programmatically send a predefined message to the server using TCP/IP sockets. The server is expected to listen on localhost (127.0.0.1) at a predefined port (e.g., 12345).

Output

The client should print to the console the message received from the server. This output should match the original message sent by the client, demonstrating a successful echo.

For the sake of example, if the client sends the message "Hello, Echo Server!", the expected output is: Received from server: Hello, Echo Server!

Method

Your solution should include both a server and a client program as follows:

- Server Program: The server should bind to the specified host and port, listen for incoming connections, accept a connection from the client, receive a message, send back the same message to the client, and then close the connection.
- Client Program: The client should connect to the server using the specified host and port, send a predefined message, wait for a response from the server, print the received message to confirm it matches the sent message, and then close the connection.

Evaluation

Submissions will be evaluated based on the following criteria:

- The server must successfully start, accept a connection, correctly receive and echo back the message received from the client, and then close the connection.
- The client must successfully connect to the server, send the predefined message, correctly receive and verify the echoed message, print the received message, and then close the connection.
- Accuracy of the implemented protocol, ensuring that the message is correctly sent, received, and echoed back.
- Correct handling of network sockets, including opening and closing connections properly.