

## Server-Client Communication (Server)

### Description

In this problem, you are asked to test the interaction between a server and a client through socket programming. The server will send a greeting message to the client, and the client should receive this message correctly.

### Input

There is no input for this problem as the interaction happens over a network connection established between the server and the client. The server will listen on localhost (127.0.0.1) at port 12345 and send a specific message to the client upon connection.

### Output

The client should output the message received from the server to the standard output. The complete output of program is as follows (you do not need to write the unit test as it is provided in the skeleton):

– Server output:

```
Test handle_client_connection ...
Got a connection from ('127.0.0.1', 12345)
send called with: call(b'Hello, Client!')
close called with: call()
Test start_server ...
Listening on 127.0.0.1:12345 ...
Got a connection from ('127.0.0.1', 12345)
accept called with: call()
bind called with: call(('127.0.0.1', 12345))
listen called with: call(1)
```

### Method

Your task is to implement and run both the server and client programs as provided. Ensure the client successfully connects to the server, receives the message, and prints the exact expected output.

– Server Program: The server should start, bind to the specified localhost and port, listen for incoming connections, accept a client connection, send the message "Hello, Client!", and then close the connection. It must handle a single client connection

before shutting down for the purpose of this problem.

- Client Program: The client should connect to the server's specified host and port, receive the message from the server, print the received message in the specified format, and then close the connection.

#### Evaluation

The submission will be evaluated based on the following criteria:

- The server must successfully start, send the correct message to the client, and close the connection.
- The client must successfully connect to the server, receive the message, and print the output in the specified format.
- The use of unit tests for both server and client to ensure the correct behavior of sending and receiving the message.