

## Code Assignment 2

## Chat Server Part 2: Server Operator

The deadline to uploading your source code to iCorsi is on the 17th of October, at 23:59. Late submission policy will apply; see the Intro slides for details. Upload a single Python project, with separate main files for each exercise. You can use the provided `template-*.py` files as a starting point. If you are using a language other than Python, include a Dockerfile that builds and runs your project.

Please take extra care to only submit source code (and not build artifacts).

### Exercise 1 - (3 points)

Spawn a thread in the server to work as the server operator. This way it will be possible to interact with the server while it is running. The operator should display the number of users connected via the command “num.users”.

To test the operator, start the server and check “num.users” when different amounts of clients are connected.

### Exercise 2 - (3 points)

The interaction between clients and servers involved only sending text messages so far, it is time to create a message format to enable the sending of structured data. We are going to use [Google's Protocol Buffers](#) to design our format. Install the library for your language of choice, and the compiler for the protocol buffer language (protoc, likely packaged for your operating system; if not, see the installation instructions on the website). In our format a normal message will be an object with the following fields:

1. `int64 from`, specifying the sender;
2. `int64 to`, specifying to whom the message is addressed;
3. `string msg`, specifying the message content.

Use `protobuf` in your project and test the client-server interaction. The server should echo the message back to the client, as done in the previous assignments.

### Exercise 3 - (4 points)

Create another message in the format file, with the goal of enabling a fast handshake. It should have the following fields:

1. `int64 id`;
2. `bool error`.

After receiving a connection, the server should assign an id to the client and send the handshake message; if some error occurs, the error field should be set to true and the connection should be closed. Make the client and the server work with the specification listed above.