Object Oriented Architectures & Secure Development

0. General information

All protocols in this session are text-based and line based.

1. First client

First, write a client to communicate with an existing server (the host and port are communicated during the lab).

The purpose of this first client is to get your personal messages shown on the large screens in classroom.

You can do this by sending the following message to the server:

```
SEND <your message>
```

Then, `<your message>` will be shown on the screens.

2. First server

Now, write a server to communicate with clients (written by you and your colleagues).

The purpose of this first server is to reveal your name (i.e., the name of the author/programmer).

You have to invent the protocol yourself, the only rules are:

- it has to be text-based and line-based;
- the client has to be able to figure out the protocol by reading the first line.

Below you find some simple examples, but you are welcome to invent one of your own.

For example:

```
Client connects
Server sends: Welcome to this server, the next line will contain my name
Server sends: Joske Vermeulen
```

```
Client connects
Server sends: Welcome to this server, I understand the following message
Server sends: GET NAME
Server sends: GET AGE
Server sends: GET ADRESS
Server sends:
Client sends: GET NAME
Server sends: Joske Vermeulen
```

```
Client connects
Server sends: Welcome to this server, tell me what 7+3 is and I will tel
Client sends: 4
Server sends: Try again: 7+3
Client sends: 10
Server sends: Joske Vermeulen
```

3. Second client

Now write a client (if you did not do it already) to communicate with **your** server.

⚠️ **Only proceed to the next phase if you finished the previous parts.**

4. Publish your own server

The first client you wrote was able to send messages to the *teacher server* using the `SEND <your message>` -command/message.

Now, you should know this server understand another type of message, which can be used to publish your server as *"Up and running"*

```
REGISTER <ip> <port>
```

Use this command to register your server.

5. Collect as many names as possible

Now it time to start a little competition.

Try to map as many ip-addresses to the names of your colleagues. You do this by connection to each of your colleagues' servers and try to retrieve their name.

To know which of colleagues have a running server you can "ask" the *teacher server* by sending the `USERS`-message.