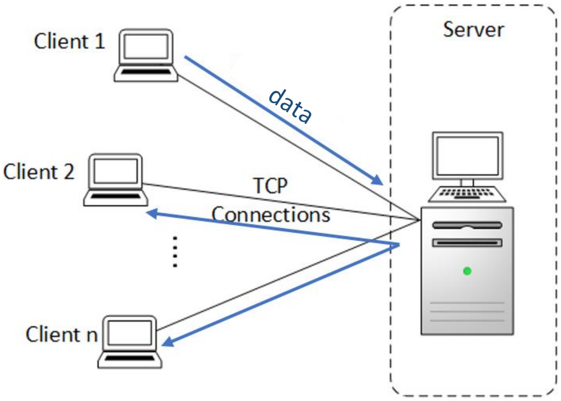
Due Date: 20.03.2019 ***27 February 2019***

BLG433E
COMPUTER COMMUNICATIONS
HOMEWORK1

OBJECTIVES

The purpose of this homework is to introduce you socket programming in python language. You will experience about creating a socket, binding a socket and port number, connection establishment with clients, listening connections, sending and receiving messages over created connections, etc. In the homework, you are expected to build a **remote quiz program** that allows multiple users to take a quiz from a single remote server. You should use multiple **TCP connections** on a client-server architecture with multiple clients. Please, prepare the program with features specified in the following section.

REMOTE QUIZ PROGRAM



The system architecture of the remote quiz program is shown with the figure above. It has two main components as a single server and multiple clients:

- **A client** communicates with the server to send an answer for the questions received. At the same time, the client should also be capable of getting questions and the final score from the server.
- **The server** may get a connection request from a new user or an answer from one of the current users simultaneously. It has the responsibility of recording connection requests, distributing the questions and evaluate a final score to all of the participants in the quiz.

The Application Logic:

A TCP connection must be established between the server and each of the clients. A username should be requested by the server as soon as a connection is established.

The server must distribute questions to the participants and receive their answer accordingly. You should provide 5 questions with multiple choice answers on the server about the topics covered in the lecture so far. The server is also responsible for calculating a final score for each of the participants.

You may consider to provide a timestamp whenever a question received for the client side. Moreover, you may also provide a timer that counts down for 1 minute to send the answer as an optional (bonus) part.

Deadlocks and starvation should be avoided among the clients. Moreover, the communication between a client and the server should be sequential once the quiz is started. In more detail, a client must not send an answer before displaying the question and similarly, the server must not accept an answer before sending a question to the relevant client.

Coding Details:

You can prepare your code using a desired version of python programming language. Do not forget to bring your computer to the demo in order to display your work properly.

You should use a threading approach for the server to provide requested application logic. You should consider any faulty cases through a scenario and handle them.

You are free to provide a console application or another GUI for testing your program as an optional (bonus) part. Moreover, it is convenient to test your source code on a single host.

Report Details:

No report files requested for the homework, since you will present the work done in the demo session on 20.03.2019.

ORGANIZING YOUR SUBMISSIONS

The homework should be prepared **groups of 2 members**. There must be at least two source file (server.py and client.py) and a readme file including information about group members. The delivery is accepted through Ninova until the submission deadline.