

# BLG 433E COMPUTER COMMUNICATIONS

## 2019-2020 FALL / PROJECT 1

### SOCKET PROGRAMMING

**Instructor:** Prof. Dr. Sema F. OKTUĞ ([oktug@itu.edu.tr](mailto:oktug@itu.edu.tr))

**Assistant:** Mertkan AKKOÇ ([akkocm@itu.edu.tr](mailto:akkocm@itu.edu.tr))

**Due on:** November 06, 2019 **till** 23:59

#### PROJECT

The purpose of this project is to enhance your experience with socket programming. You will experience about creating sockets, binding a socket and a port number, listening connections, sending and receiving messages over created connections etc. In the project, you will create a server that supports multi-player “**Hangman**” game with two or more clients. The project is explained below. You can use “Python” or “C” to write server and client programs.

#### PROJECT DESCRIPTION

**Server should do the basic things below:**

1. Asking that how many players there will be. This will be just once.
2. Registering each player/client to the server so each player will have a unique ID. For registration, each player creates an account for itself on the server (a password is necessary for each account).
3. Communicating with just one player or more than one player at a time.
4. Keeping latest status information and showing this information to the players.
5. Getting guesses from players in order and compare with the phrase.
6. Keeping the order of players and inform to all players about which player will play next (order will be determined from connection order to the server to play).
7. Keeping wrong guesses for both letters and phrases. These guesses should be seen on the screens of the players.
8. Showing current guess of the player to other players.
9. Calculation of remaining allowed attempts.

**Client** is an interface between the users and the server. It should be as user friendly as possible.

**Latest Status Information** should have last version of phrase with known and unknown parts, remaining attempts for guessing.

**Rules:**

1. The game starts after specified number of players connect to the server.
2. Allowed number of attempts decrease one by one in exchange for each wrong letter guess or each wrong phrase guess.
3. Server accepts lowercase or uppercase letter for both letter guesses and phrase guesses. But if user enters an uppercase letter, you have to convert to lowercase.
4. If the user enters two or more letters instead of one letter or enters a number, these count as wrong guess and as a result number of attempt decrease.
5. The game terminates after 7 wrong guesses.
6. The game also terminates if one player guesses whole phrase right or all letters are guesses correctly.
7. The player who registered before should inform the server if she/he wants to play again.

**GRADING:**

1. Registration and keeping registration information is 10 points.
2. Communication among the server and **two or more** clients at a time is 20 points.
3. Communication between the server and **a** client in the right order is 20 points.
4. Keeping the wrong guesses for both letter and phrase is 5 points.
5. Keeping and calculation of remaining number of attempts is 5 points.
6. Comparing phrase guesses with the actual phrase is 5 points.
7. Comparing letter guesses with the actual phrase is 5 points.
8. **Report** is 30 points.

**!!! REPORT** includes screenshots that show messages among server and clients and also includes explanation of your server program and client program. In the explanation, please explain which part of your program does which job. If you add modules or library, please explain why you add that module or library.

**SUBMISSION:**

The project is built **individually**. All relevant code (server and client programs) and text files that you need to create, should be under one directory (If you are coding on python3, please add also codes that can be executed with python3). This directory should be named with the number of the student such an example: "150100000.zip". It should be uploaded to Ninova **till November 06, 2019 / 23:59**.

HAVE FUN ☺