# Computer Networking
# Coursework Assignment

### Birkbeck, University of London

### Academic Year 2018/19

The assignment requires two programs (client and server) in Java or Python, that implement the game described below.

## Client

The client has to be coded according to the following specification.

- At startup, the client prompts for a string $S$ from the user.

- The client then splits the string $S$, assumed to be of length $n$, into two strings $S_1$ and $S_2$ that are of equal length $n/2$ if $n$ is even, or of lengths $(n+1)/2$ and $(n-1)/2$ respectively if $n$ is odd.

- The client then:

  1. sends $S_1$ and $S_2$ to the server in two distinct UDP packets;
  2. waits for a string $R$ from the server, in another UDP packet;
  3. checks if $R$ is equal to the concatenation of $S_2$ and $S_1$;
  4. displays the result of the final test to the user.

*Example:* The user enters $S$ as `perfume`; the client sends $S_1$ and $S_2$ as `perf` and `ume` respectively; then it receives $R$ as `umeperf` from the server and verifies that $R$ is the concatenation $S_2 S_1$.

## Server

The server has to be coded according to the following specification.

- The server constantly waits for UDP messages.

- The server waits for *a sequence of two distinct UDP messages* containing strings $S_1$ and $S_2$.

- The server then:

  1. receives $S_1$ followed by $S_2$ in two different UDP packets;
  2. constructs $R$ as the concatenation of $S_2$ followed by $S_1$;
  3. sends $R$ to the client as a UDP packet;
  4. starts again.

*Example:* the server receives $S_1$ and $S_2$ as `caulif` and `lower`, constructs $R$ as `lowercaulif` and then sends $R$ to the client.

## Presentation

The student has to produce the result of the coursework as follows.

1. The Java or Python code for the client and the server, as above specified, is to be produced.

2. The student will produce a brief, informal description of the code (PDF file; *no Microsoft Word or other formats*). The description should include screenshots of two executions of a client/server pair. An actual image has to be produced as a screenshot; one screenshot for each pair (two in total) will be sufficient (separate screenshots for each window are of course allowed).

3. The screenshots, accompanied by a brief explanation, will have to show a window for the client and one for the server; the client and the server will be coded so as to display suitable messages showing the execution of each step.

All the code and documentation (two source files for client and server plus a PDF file) will have to be uploaded in a *Zip* file named according to the following template: *surname_name_id.*zip.

*The deadline for the submission is the 28th of April, 2019, at 23:55 London time (GMT+1).* All late submissions, even if they are late by a very short time, will be capped at 40, unless a Mitigating Circumstances claim is approved. Students are strongly encouraged not to submit at the very last minute, so as to avoid technical problems which may cause a late submission.