

## Problem Set 1

### 1 Instructions

Create a folder on your computer named **ps1\_lastnamefirstname**. You will include all files required for this assignment in this folder. When you are ready to turn-in your assignment, you will compress (ZIP UP) your folder and submit the ZIP file to the 'Assignments' area on Sakai. Please don't email me your submission because CMC's email server blocks emails with ZIP file attachments.

### 2 Part 1

For part 1, you will complete the provide template to create a Tic Tac Toe game implemented in Java to exchange messages between the client and server using sockets.

The client server paradigm works as follows:

- A server is started waiting for a client(s) to connect.
- A client connects
- Server and client exchange information (via the defined protocol).
- The client and the server close their connection.

Here is some general information. Hosts have an IP address and a port (numbered from 0 - 65535). Servers listen on a specific port. Some port numbers are reserved so you can't use them when you write your own server. Multiple clients can be communicate with a server on a given port (they must specify the IP address of the server to connect to). Each client connection is assigned a separate socket on that port. Client applications get a port and a socket on the client machine when they connect successfully with a server.

For this part of the assignment, we want to write a server/client application to simulate a tic tac toe game. The server will wait for two clients to connect and maintain the game board. The clients will connect to the server, listen for instructions, and specify their moves.

The server and the client(s) communicate through a simple text protocol'.

*Server:*

- The server creates a Game object which will maintain the state of the board.
- It will wait for two clients to connect, player X and player O.
- Once the clients connect, we launch a thread for each player to manage the messages send/received from the client.

- Your task is to complete the method `run(.)` in class `Player`. The function loops to receive messages from the client. If the client makes a move, we should ensure the move is legal and send the appropriate message.

*Client:*

- The client is implemented as a thread. This thread will listen for messages from the server.
- The first message will be a 'WELCOME' message in which we receive our mark(X or O).
- Then we go into a loop listening for 'YOUR\_MOVE', 'VALID\_MOVE', 'OPPONENT\_MOVED', 'VICTORY', 'DEFEAT', 'TIE', 'OPPONENT\_QUIT' or 'MESSAGE' messages, and handling each message appropriately.
- The 'VICTORY', 'DEFEAT' and 'TIE' ask the user whether or not to play another game. If the answer is no, the loop is exited and the server is sent a 'QUIT' message. If an 'OPPONENT\_QUIT' message is received then the loop will exit and the server will be sent a 'QUIT' message also.

Feel free to modify any part of the code you feel is necessary. I have basically included 'TODO' statements to indicate the sections where you should fill-in the code. The basic structure of the program and the server/client communication is given to you. Do spend time understanding the code even though the template is given to you so that you can tackle more challenging problems in the future from scratch.

### 3 Part 2

For part 2, you will use HTML tags and CSS style selectors to create a webpage of your choice. First pick an area of your own interest, such as your summer vacation, internship, surfing, birds, airplanes, travel, Mark Twain, swimming, etc. The content of your document will depend on your imagination and creativity. You can start off providing a brief introduction of yourself, and then include information about your chosen topic.

I know that many of you may already have webpages, but please create a new site for this part of the assignment.

This problem set requires that you build the HTML and CSS files from scratch. You may look at how other people organize their websites / homepages for inspiration, but please do NOT use any existing templates from the web (using bootstrap is okay). As you write your document, you will want to use features available in HTML and CSS to make your page more readable and appealing.

**I expect to see at least the following HTML features and tags in your submitted files:**

- Include webpage title in the heading
- Include webpage information using the meta tag in the heading
- Include comments in your HTML and CSS files

- Design your HTML page layout using the tags: header, footer, section, article, nav
- Include a ordered or unordered list: ol and ul tags
- Include a table : table, th, tr, td tags
- Include a few headings: h1, h2, h3, h4, h5, h6 tags
- Include hyperlinks : a tag
- Include images (using img tag) and update some attributes like (alt, width, height, border, etc.)
- Include 1 video using either the iframe or video tag (this can be a video on youtube)

**For CSS, I expect you to include:**

- All three ways (inline, internal, and external) of using CSS that we studied in class.
- Usage of class (using dot) and id ( hashtag) selectors to apply styling in more specific ways
- Different panels / sections in your html document (like a heading, footer, side-panels, etc.) each with its own CSS styling information.

Your submission for part 1 should include ps1-1.html and ps1-1.css and any additional html files, images, videos, audio files, etc.

## 4 Submission

- Your final submission should be a ZIP file that consists of part 1 and part 2 files.
- It is your responsibility to verify that the ZIP file was uploaded correctly, and contains all your files. Double-check what you upload just in case!
- Be sure to VALIDATE your HTML before submission!!!!
- Part 1 will be graded on correctness. While Part 2 will be graded based on completeness (including required features) and the effort shown in the submission.