

EE4216 Test 2 (2020/21 Sem A) Duration: 120 minutes

Important Notes:

- This paper has 2 questions.
- To answer these questions, download the project template from Canvas. Write your answer/code in the template files provided to you.
- Rename the project folder to **test2-xxxxxxx** where the suffix is your 8-digit student ID.
- Compress the project in ZIP format and submit it to Canvas at the end of the test.
- DON'T alter the project structure and configuration file.
- DON'T use any other libraries unless specified in the question.

Question 1 (30 marks)

You are asked to write a simple number guessing game. The game is entirely implemented with a single Java servlet class. Implement the class Q1.java in the project template. The game flow is described with screenshots below.

Pick a number between 1 to 10. You have 4 chance(s) left. Submit No guess is made.	The game starts when the servlet page is loaded. A random secret number will be generated between 1 and 10 and be stored in the session. The player has 4 chances to guess the secret number. A guess can be submitted with the HTML form. The no. of chances is also stored in the session. When the servlet page has no input for guessing the number, the message “No guess is made” is shown.
Pick a number between 1 to 10. You have 4 chance(s) left. Submit Invalid input.	When an empty or a non-numerical string is received, the message “Invalid input” is shown. Note that the unused chance remains unchanged if no input or invalid input is received.
Pick a number between 1 to 10. You have 3 chance(s) left. Submit Your guess, 5, is too small.	Subsequent guesses will be checked against the secret number in the session and a hint, either <u>too big</u> or <u>too small</u>, will be provided. Each valid guess consumes one chance no matter it is correct or not.
Pick a number between 1 to 10. You have 2 chance(s) left. Submit Bingo! My secret number is 7. Play Again	The game continues until the player hits the secret number and the bingo message will be shown. At this point, the game is reset by clearing the session. Clicking the link “Play Again” will reload the servlet page and start a new game.
Pick a number between 1 to 10. You have 0 chance(s) left. Submit You lose. My secret number is 4. Play Again	Similarly, the game ends and is reset when the player uses up all 4 chances. When the player makes a wrong guess at the last turn, an ending message with the “Play Again” link will be shown instead of the hint.

Question 2 (30 marks)

You are asked to use Vue.js to re-write the number guessing game described in Question 1. Implement the file Q2.html and Q2.js in the project template respectively.

This time, the game is event-driven and entirely implemented on the client-side. The secret random number, the number of chances as well as other custom game states are all stored in the Vue instance. You do not need to persist the data to local storage or server session.

You should implement the same game flow as illustrated in Question 1. The only exception is that you do not send the input value to a server but validate the input value in client-side when the submit button is pressed.

Similarly, when the “Play Again” link is clicked, you reset the game by resetting the game states in the Vue instance. The entire game flow is controlled under the same Vue instance lifecycle. You should not refresh or reload the page.

Note that you are required to use Vue’s native data binding approach to manipulate the DOM. Any imperative JavaScript code attempting to update the DOM will result in deduction of marks

END