EECS 1012: LAB #7 –JavaScript and HTML forms (Nov 12-16, 2017)

1. Read the lab instructions in this document and take the pre-lab quiz for Lab #7 -- the TAs will not mark your lab if you do not show them you have not scored 100% on the pre-lab quiz.

Please have a look at the examples posted here for JavaScript and HTML Forms. These examples come directly from the lecture notes.

2. GOALS & OUTCOMES FOR THIS LAB

- To learn and apply more JavaScript programming
- To learn how to use regular expressions (regex)
- To learn how to validate a simple form

3. LAB 7– TASK

[Task 1] – You will need to modify task1.html and task1.js. This focuses on regular expressions.

[Task 2] – You will need to modify task2.html, task2.css, and task2.js.

Also see accompanying video to show behavior of your JavaScript code.

4. SUBMISSIONS

1) [Manual verification by a TA]
As with the previous labs, when you have completed all tasks, ask the TA to come and verify your code and output.

2) Moodle submission
You will see an assignment submission link for Lab6 on Moodle.

1) Create a **folder** named "**Lab7**" and copy all of your files into it. Compress this file and upload the compressed file to Moodle. To upload, please follow the instructions in the following video that we used for Lab 1:
https://www.youtube.com/watch?v=stEOh6ntV5o

TASK 1. Modify both your task1.html and task1.js to implement the regular expressions described for each example in the box. Write you regular expression in the HTML code as well in the corresponding JS. If the user enters in a string, when the button is clicked it should evaluate yours regular expression and return a true or false.

Your Name

Example 1

Regular Expression : [**Replace with your regex**]

Result:

Press to evaluate your string

Example 2

Regular Expression : [**Replace with your regex**]

Result:

Press to evaluate your string

Example 3

Regular Expression : [**Replace with your regex**]

Result:

Press to evaluate your string

Example 4

Regular Expression :[**Replace with your regex**]

Result:

Press to evaluate your string

Example 5

Regular Expression : [**Replace with your regex**]

Result:

Press to evaluate your string

Example 1: Number and spaces
Your regex should allows at least one or more number character. You can also allow spaces. Examples:
"123", "12 34", "11111 0"

Example 2: Your regular expression must start with one or more numbers and end with one ore more letters (ignore case). Example:
"1234Hello", "1B", "1234B", "1ABCD"

Example 3: Your regular expression must start with the letter S, then followed by exactly 4 numbers, then end with any uppercase letter. Example:
"S1234Z", "S9876A", "S1010E"

Example 4: Make a regular expression for Ontario license plate. It must start with for uppercase letters, then followed by 3 digits. Example:
"AYRV178", "AJAL506", "AAAA000"

Example 5: Make a regular expression to find **floating point numbers**. It must start with one or more numbers, then a period, followed by at least one or more numbers.

To include a period in your regex, use \. (as we do with a space).

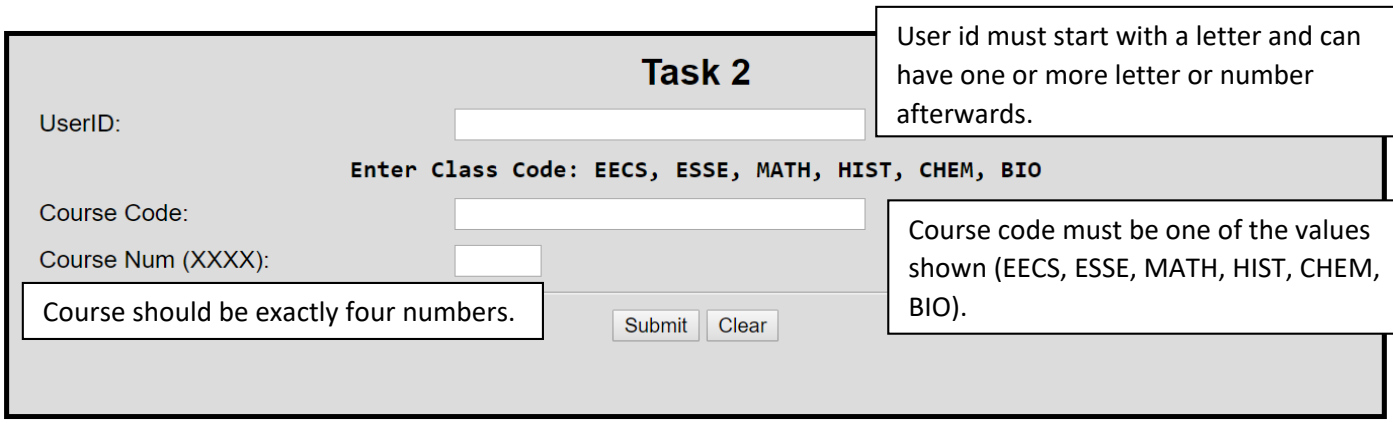
Example:

"1.0", "0.1020", "101.99", "10000.00"

Task 2. This task is a simple version of the lecture on JavaScript and HTML Forms. Modify the JavasScript file to validate your form before submitting. The HTML file is already completed for you – please look at it carefully to see the IDs associated with the fields and the layout of the HTML content. It shouldn't be necessary to modify the HTML file, but it is OK if you do. Your JavaScript should validate the form as the user types in the input (i.e. observe the "keyup" event). See below for the allowed inputs.

If an input is correct or incorrect, change the element next to the input to say either "CORRECT" or "INCORRECT".

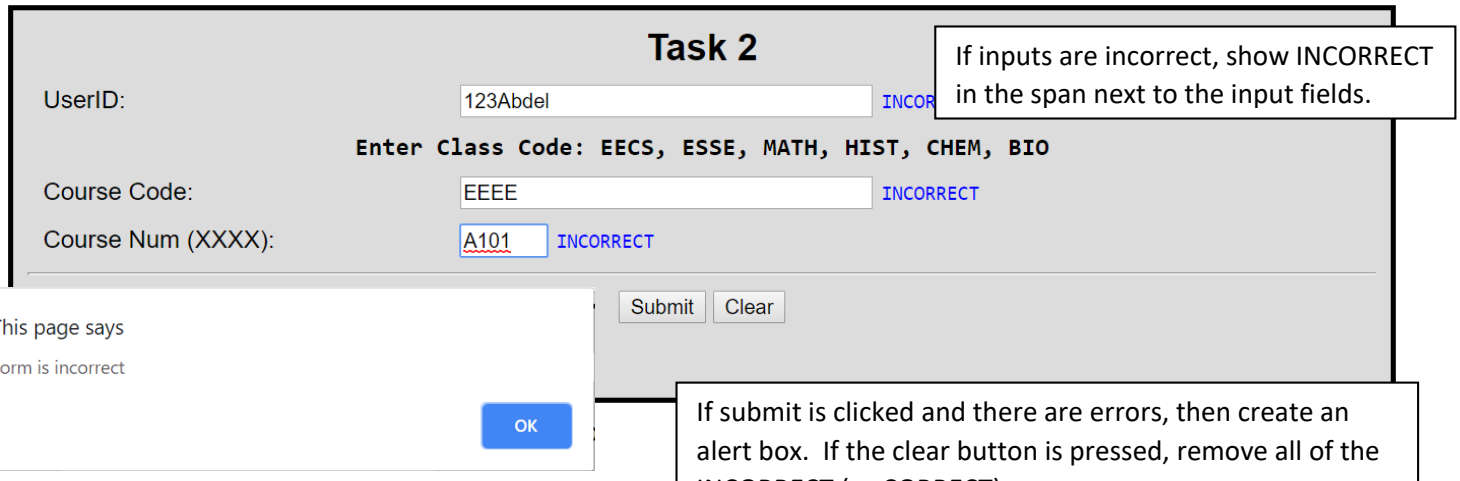
If the user clicks "Submit" and all the inputs are correct, submit the form (see Event lecture). If any of the inputs are "INCORRECT", have an alert window say there is an error. If the "clear" button is clicked, you should also clear all the errors beside the input fields.



User id must start with a letter and can have one or more letter or number afterwards.

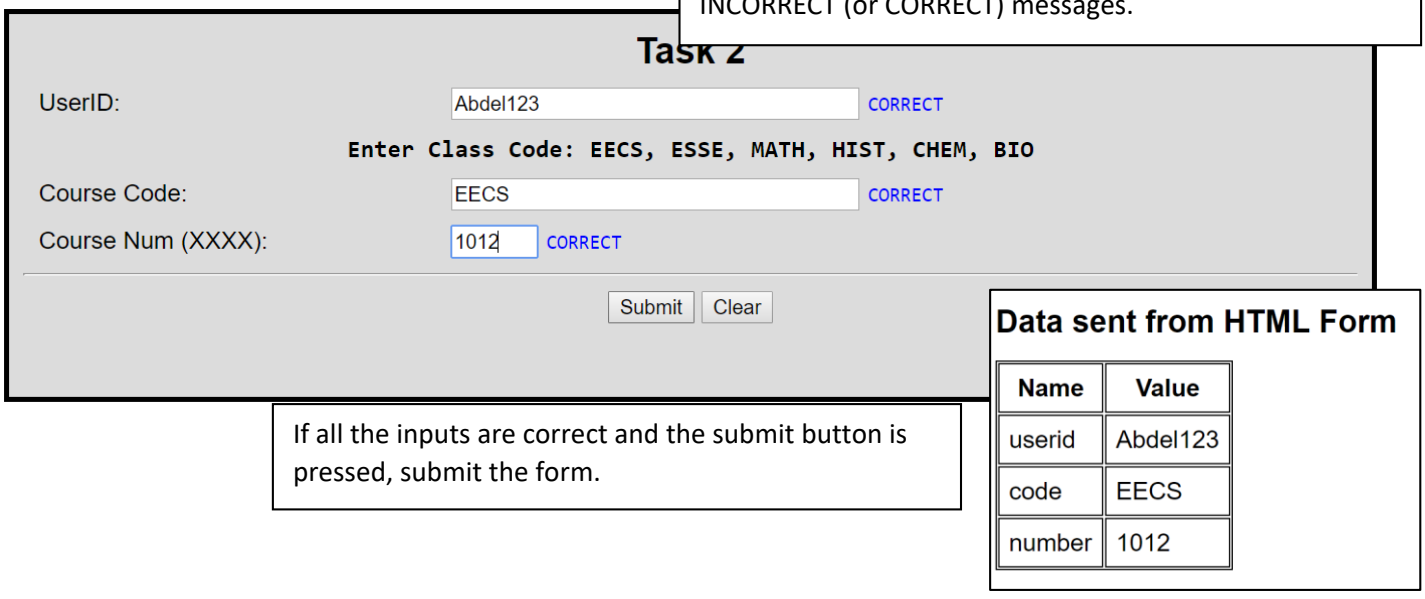
Course code must be one of the values shown (EECS, ESSE, MATH, HIST, CHEM, BIO).

Course should be exactly four numbers.



If inputs are incorrect, show INCORRECT in the span next to the input fields.

If submit is clicked and there are errors, then create an alert box. If the clear button is pressed, remove all of the INCORRECT (or CORRECT) messages.



If all the inputs are correct and the submit button is pressed, submit the form.