**Practical 3**

**Part 1: Knowledge Check**

**Question 1**

<script src=“scripts/jquery-1.9.2.js”></script>

<script src=“scripts/jquery-1.9.2.js”>

Based on the code snippet given above, are these two statements the same?

* These two statements are having the same purpose/effect.
* However, some browsers do not handle “self-closing” <script> tags correctly.
* Therefore, some browsers fail to load the script.
* To ensure proper loading of the script, always use an explicit close tag.

**Question 2**

What are the pros and cons of including jQuery at the top of the page versus bottom?

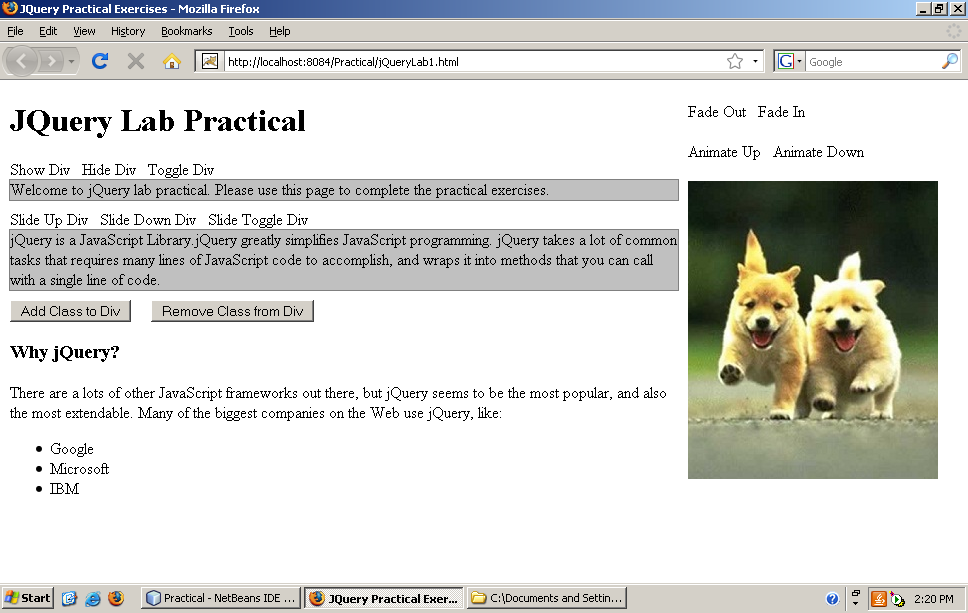
* It’s common to place the <script> element in the <head>tag of a document.
  + Structurally, it is a good location as it separates the behavior (jQuery) from the page content (HTML).
  + Functionally, this can be a problem whereas the browser might blocks (pauses) the script to load before it continues loading and rendering the rest of the document.
* Placing the <script>element(s) just before the </body> close tag will be able to increase the page responsiveness.
  + This causes the browser to load and render all other page content before loading and executing the jQuery code.
  + Therefore, it will be able to give users a functional web page before the interactive components implemented in jQuery are fully loaded and initialized.

**Part 2: Lab Exercises**

**NOTE:** You are required to get the necessary practical files from the respective instructor to complete the following lab exercises.

**Question 1**

Open the project folder named **P3** andlocates the following page (**jQueryLab1.html**):



**Figure 1: GUI for jQuery Lab Practical 1**

**Instructions:**

With reference to Figure 1, complete the exercises by fulfilling the following requirements:

1. Modify the <script> element to link to the local jQuery library file.
   1. NOTE: Google hosted jQuery library file is not workable in the computer lab as it required internet connection.
2. Add a second <script> element to the <head> element of the page.
3. Add the Document Ready Event to the <script> element created in step 2.
4. In the Document Ready Event, add a click event for the anchor element that assigned with **showDiv** identifier. In the function, add the jQuery code to show the **intro** division.
5. In the Document Ready Event, add a click event for the anchor element that assigned with **hideDiv** identifier. In the function, add a jQuery code to hide the **intro** division.
6. In the Document Ready Event, add a click event for the anchor element that assigned with **toggleDiv** identifier. In the function, add the jQuery code to toggle the **intro**division.
7. In the Document Ready Event, add a click event for the anchor element that assigned with **slideUp** identifier. In the function, add the jQuery code to slide up the **about** division.
8. In the Document Ready Event, add a click event for the anchor element that assigned with **slideDown** identifier. In the function, add the jQuery code to slide down the **about** division.
9. In the Document Ready Event, add a click event for the anchor element that assigned with **slideToggle** identifier. In the function, add the jQuery code to slide toggle the **about** division.
10. In the Document Ready Event, add a click event for the button element that assigned with **addClass** identifier. In the function, add the jQuery code to add the “red” class to the **why** division.
11. In the Document Ready Event, add a click event for the button element that assigned with **removeClass** identifier. In the function, add the jQuery code to remove the “red” class from the **why** division.
12. In the Document Ready Event, add a click event for the anchor element that assigned with **fadeOut** identifier. In the function, add the jQuery code to fade out the dog image.
13. In the Document Ready Event, add a click event for the anchor element that assigned with **fadeIn** identifier. In the function, add the jQuery code to fade in the dog image.
14. In the Document Ready Event, add a click event for the anchor element that assigned with **animateDown** identifier. In the function, add the jQuery code to animate the dog image by changing the height value to 400px.
15. In the Document Ready Event, add a click event for the anchor element that assigned with **animateUp** identifier. In the function, add the jQuery code to animate the dog image by changing the top value to 100px.

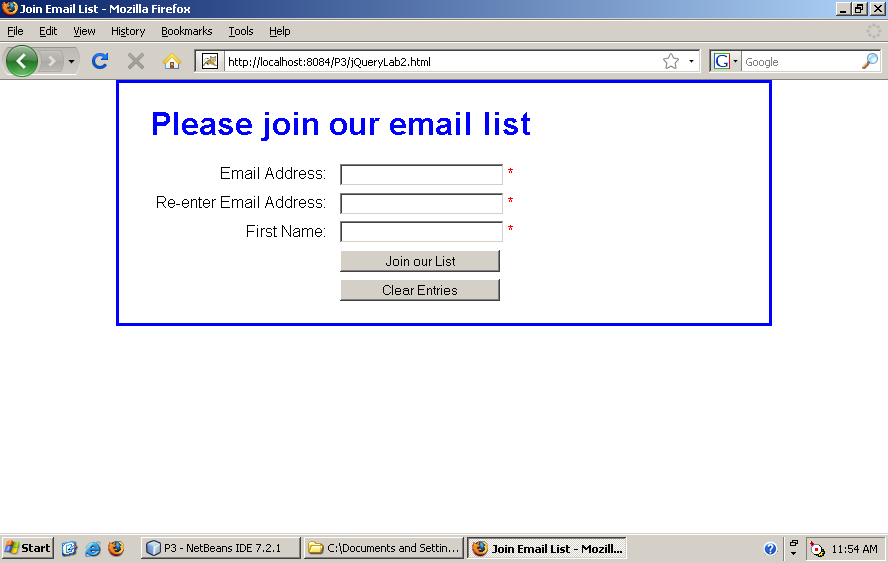
**Question 2**

Open the web page named **jQueryLab2.html** which located under **P3** folder.

In this practical, we will host the jQuery code in a separated Java Script file.

**Instructions:**

With reference to Figure 2, complete the exercise by fulfilling the following requirements:



**Figure 2: GUI for jQuery Lab Practical 2**

1. Create a new Java Script file named **JQueryLab2.js** (In this practical, we will host the jQuery code in a separated Java Script file and make it located under scripts folder).
2. Link the newly created Java Script file to the HTML page.
3. In the new Java Script file, add the Document Read Event to ensure that to page is ready before loading the jQuery Code.
4. Add a click event for the Join our List button. The click event contains several features which include:
   1. Validation
      * Required Fields: Email Address, Re-enter Email Address and First Name. Users are not allowed to leave the fields empty.
      * Validate the fields and display an error message (eg: “This field is required.”) if the users blank out the required fields.
      * Ensure that Email Address and Re-enter Email Address fields contain the same value.
      * Validate these fields and display an error message (eg: The email address entered must be the same.”) if the value entered by the users are different.
   2. Navigation
      * Navigate the page to jQueryLab2\_join.html if all entries are valid.
   3. Additional Enhancement
      * Change the field’s background color to #FFF8C6 if the user focuses on a particular field.
      * Change the field’s background color to #FFFFFF if the user left the field.
      * When the page is loaded, always place focus on the first text box.
5. Test the changes to ensure that the program is work as intended before you proceed to the next step.
6. Add a click event for the Clear Entries button. The click event contains several features which include:
   1. Clear all the text boxes by setting the value to an empty string (“”).
      * This feature can be achieved using one statement.
   2. Put the asterisks (\*) back to the span elements that are displayed at the right hand side of the text boxes to show that entries are required.
      * This feature can be achieved using one statement.
   3. Add one more statement to this event handler that move focus to the first text box.
7. Add another double-click event handler to this program. When the user double-clicks on any of the text boxes, the event should do the same thing that the Clear Entries button’s event does.
   1. Hint: The easiest way to do that is to trigger the click event of the Clear Entries button from the handler for the double-click event.
8. Comment out the line of code that you jut used to trigger the click event of the Clear Entries button. Then add a statement to the double-click event handler that only clears the text from the text box that the user double-clicks in.

**Question 3**

Open the web page named **jQueryLab3.html** which located under **P3** folder.

In this practical, we will host the jQuery code in a separated Java Script file.

**Instructions:**

With reference to Figure 3, complete the exercise by fulfilling the following requirements:



**Figure 3: GUI for jQuery Lab Practical 3**

1. Create a new Java Script file named **JQueryLab3.js** (In this practical, we will host the jQuery code in a separated Java Script file and make it located under scripts folder).
2. Link the newly created Java Script file to the HTML page.
3. In the new Java Script file, add the Document Read Event to ensure that to page is ready before loading the jQuery Code.
4. Observe the code given in the HTML page, you will realize that the <img> elements contain two different image URLs, one in src attribute another in id attribute. By default it displaying the image from src attribute in <img> element as illustrated in Figure 3.
5. Enhance this program to make it more interactive by adding a mouse over event handler. Change the image using the image URL from id attribute in <img> element when the user mouse over the image.
6. Insert another mouse out event handler to change the image back to the original image using the image URL from src attribute in <img> element.
7. **Hint:** 
   1. Loop the 3 images using jQuery .each function and save the src and id attribute into a variable. When mouse over or mouse out the image, change the picture through the mouseover and mouseout event handler by manipulating the src attribute value using the variable saved.
   2. If you are having problem in applying jQuery .each() function, consult the jQuery API documentation to complete the exercise. The jQuery library is large; learning to find answers in the documentation is an important part of the process.

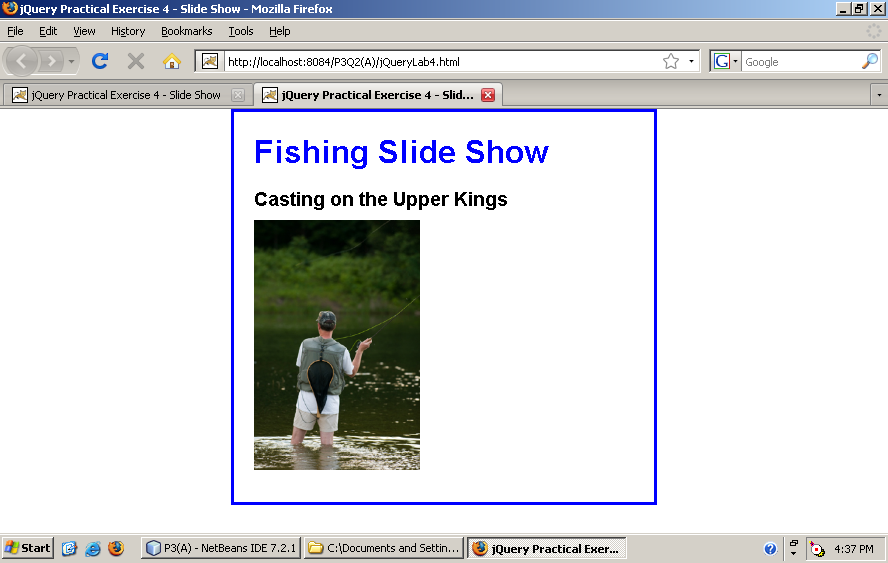
**Question 4**

Open the web page named **jQueryLab4.html** which located under **P3** folder.

This question required skills in Java Script (eg: setInterval, Timer, etc) and jQuery (hide, show, etc).

**Instructions:**

With reference to Figure 4, complete the exercise by fulfilling the following requirements:

****

**Image Caption**

**Figure 4: GUI for jQuery Lab Practical 4**

1. AHTML page as illustrated in Figure 4 has been given. The page contains five different <img> elements in the <div> tag and only the first image can be displayed statically. Therefore, you are required to improve the program by modifying the HTML page and put in the animation effect using jQuery code so that it will be able to work like a simple animated slide show.
2. Others than animating the image, display the image caption when the image slide up and then back down as the show moves from one slide to the next. Use the value in alt attribute from <img> element as the caption. (**Hint:** Do not hard code the caption; manipulate the <img> attribute to retrieve the alt value.)
3. Loop the image as to create a continuous running slide slow.
4. Experiment with different effects and durations until you get the slide show that work in a way that you like.

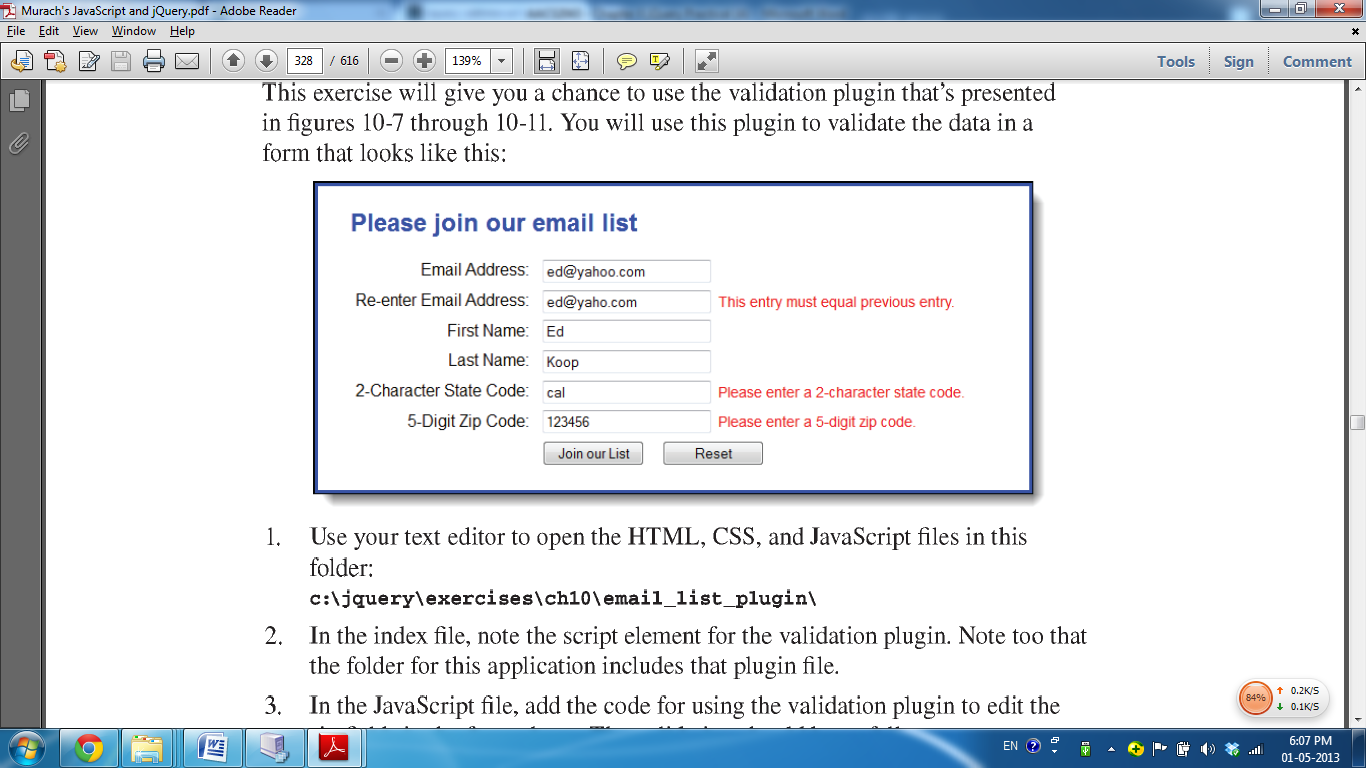
**Question 5**

Open the web page named **jQueryLab5.html** which located under **P3** folder.

This exercise will give you a chance to experience the validation plugin.

**Instructions:**

With reference to Figure 5, complete the exercise by fulfilling the following requirements:

****

**Figure 5: GUI for jQuery Lab Practical 5**

1. Create a new Java Script file named **JQueryLab5.js** (In this practical, we will host the jQuery code in a separated Java Script file and make it located under scripts folder).
2. Link the newly created Java Script file to the HTML page.
3. Add in another <script> tag to link the form validation plugin file named jquery.validate.min.js that located under scripts folder to the <head> element of the HTML document.
4. In the new Java Script file, add the Document Read Event to ensure that to page is ready before loading the jQuery Code.
5. When the page is loaded, place focus on the first text box.
6. After that, add code to use the validation plugin in order to validate the input entered into the six fields as shown in Figure 5. The validation should be as follows:
   1. Allof the fields are required.
   2. The first two fields should be valid email addresses and the second email address should be the same as the first.
   3. The state code should be a two-character code.
   4. The zip code should be a five-digit number, so it should be tested both for digits and characters.
7. **Hint:** jQuery validation plugin uses the following code format:

[$](http://docs.jquery.com/Core/$)("#formID").[validate](http://docs.jquery.com/Plugins/Validation/validate)({

rules: {

field1ID: {

//validation rules/setting

//Example: set as required field

**required: true**,

…

…

}

Field2ID:{

…

…

}

}

});

1. **Hint:** Complete the exercise by referring to the jQuery validation plugin documentation. You can get the plugin documentation from http://docs.jquery.com/Plugins/validation

**Question 6 (Bring home exercise)**

**\*\* Freestyle, no fix answer. You can ask students to form a group of 2 to complete this exercise and award them with marks based on their work.**

Create an online assessment page (eg: online test) that implementing jQuery UI. You are required to apply and demonstrate all the UI interaction (draggable and droppable, resizable, selectable and sortable) into the page.

Complete the exercise by fulfilling the following requirements:

* Your online assessment should contain minimum 5 questions in any subjects/ courses that having the jQuery UI interaction ability.
  + For example, you may include a drag and drop questions which allow the user to drag on a particular box or picture and then drop it on the correct location/answer (You can ignore the correctness of the answer provided by user in this exercise).
* Include a datepicker widget into the exercise so that the user can choose the assessment date.
* Finally, when the user click on the “Submit” button to complete the assessment, prompt a dialog box to ask the user’s response on the confirmation for the submission.
* You may add in any others relevant widgets/ jQuery UI components to improve the page performance or usability.
* Express your creativity in designing the page to make it more interactive.
* **Hint:** Visit jQuery UI official site (<http://jqueryui.com/>) if you are having problem with jQuery UI.