1. [20] Create a new page called images2.html. Use a table to nicely arrange two text fields, one for inputting an image URL and one for inputting an image caption, along with text indicating the purpose of each text field, a button with label “Add Image”, and a button with label “Show Random Image”.

2. [10] Inside a <script> element, define an AnnotatedImage object using the following code:

   ```javascript
   function AnnotatedImage(url, caption) {
       this.url     = url;
       this.caption = caption;
   }
   ```

3. [10] Also inside a <script> element, Create a new instance of an Array using code similar to the following:

   ```javascript
   var annotatedImageArray = new Array();
   ```

4. [30] When the “Add Image” button is clicked, create a new instance of AnnotatedImage using the url and caption specified in the text fields. Add this instance to your array using the Array push() method. For example, considering an Array named “annotatedImageArray” and an AnnotatedImage instanced named “image”, you would add the AnnotatedImage instance to the Array using:

   ```javascript
   annotatedImageArray.push(image);
   ```

The text fields should also be cleared as a result of clicking this button.

5. [30] When the “Show Random Image” button is clicked, the page should display one of the images that have been added along with its caption. Use the Math.random() and Math.floor() functions to select a random AnnotatedImage instance from the Array to display. Only one image and caption should be displayed at a time. If the Array is empty, clicking this button should have no effect (i.e., should not cause an error); you can detect this case by testing to see if the length of the Array is 0.

6. [+10] (Extra credit) Modify your page to add a “Show Next Image” button. When clicked, the next AnnotatedImage in the Array should be shown. If the last AnnotatedImage is currently being displayed, show the first AnnotatedImage. Clicking the button before images have been added should have no effect (i.e., should not cause an error). Hint: the easiest way to implement the button behavior is to keep track of index of the current AnnotatedImage. You can define an index variable and give it an initial value of -1 (so that 0 is the first index used),
similar to the way that you defined the Array, then modify the variable whenever a different AnnotatedImage is shown.

Upload your files to Blackboard (under Course Documents -> Assignment 3).