

## Course 12-780

### Advanced Python and Web Prototyping for Infrastructure Systems

Fall 2016

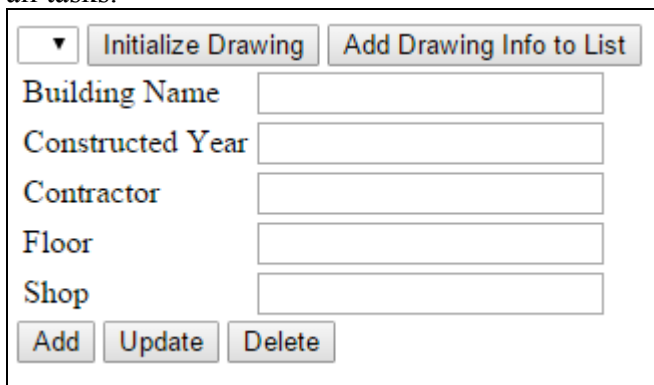
#### Assignment 2 (Total: 8 points)

**Due: Tuesday, Nov. 15<sup>th</sup>, by Blackboard by end of the day.** For each day the assignment is late, a penalty of 10% of your total grade will be deducted.

##### Task 1 (8 points) *Drawing Management In Javascript*

Please download the Task1.html and drawingManagement.js file from the assignment and save them into the same folder.

The final interface of the Task1.html should look like the following image after finishing all tasks.



The screenshot shows a web form with a dropdown menu at the top left. Below it are five text input fields labeled 'Building Name', 'Constructed Year', 'Contractor', 'Floor', and 'Shop'. At the bottom of the form are three buttons: 'Add', 'Update', and 'Delete'.

- a. There are already HTML codes for a `<select>` tag, which is a dropdown list, and labels and text boxes in the Task1.html. The five buttons are missing. Please add these buttons directly in the Task1.html page, and link the mouse click event for each of them with the following functions defined in the drawingManagement.js file (1 pt):

Button (Initialize Drawing): `function initDrawings()`

Button (Add Drawing Info to List): `function addDrawingInfoToList ()`

Button (Add): `function addDrawing()`

Button (Update): `function updateDrawing()`

Button (Delete): `function deleteDrawing()`

- b. The function `initDrawings()` created 4 drawing objects using the prototype Drawing. However, we found that one important attribute, Shop, was missing in the prototype Drawing. Please add this attribute **to the** prototype Drawing **inside** the function `initDrawings()`, and then create 2 more drawings objects with the following information (1 pt):

**Note:** Please don't modify the construction function Drawing() to add the new attribute. Add it directly to the prototype inside function initDrawings().

DrawingID	Building Name	Constructed Year	Contractor	Floor	Shop
NSH_001	Newell Simon Hall	1988	Turner	3	Architecture
NSH_002	Newell Simon Hall	1988	Turner	2	Structure

- c. The function addDrawingInfoToList() provides an example of creating HTML tag <option> in the tag <select> to fill in data for the dropdown list. Most of the codes for this function are written, except the part that assigns the DrawingID of a drawing object to the newly created <option> tag. Please complete this function (only need one line of statement) (0.5 pt).
- d. The dropdown list object, which is the HTML <select> tag, has an event *onchange*, which is triggered when the item selected in the dropdown list changes. This event is linked with the function drawingInfoListChange(). This function uses the currently selected drawing object to fill-in information in the five <input> tags (buidlingNameInput, constructedYearInput, etc.). The codes for getting the selected drawing object is already written. Please complete the codes to use information in the variable selectedDrawing to fill-in the five <input> tags (1.5 pt).
- e. The function addDrawing() is triggered by the button Add. It will first asks user to enter a string as the ID of a new drawing. Please finish the following steps (2 pt):

i. Use a dialog from Javascript to ask user to enter a new ID for the drawing object, as shown in the screenshot below;

The screenshot shows a web form with the following elements:

- A dropdown menu with "Port\_001" selected.
- Buttons: "Initialize Drawing" and "Add Drawing Info to List".
- Form fields: "Building Name" (containing "Porter Hall"), "Constructed Year" (containing "2009"), and "Shop".
- An "Add" button.
- A JavaScript dialog box with the title "JavaScript" and a close button (X).
- The dialog box contains the text: "Please enter a new ID for the drawing".
- The dialog box has an input field for the ID.
- At the bottom of the dialog box, there is a checkbox labeled "Prevent this page from creating additional dialogs." and two buttons: "OK" and "Cancel".

ii. If the user directly clicks cancel button in the input dialog, it will show a

message dialog to remind the user first enter the new ID;

iii. If the new ID is entered, this function will check all existing drawing objects to see whether the new ID already exists in the drawing list or not. If the new ID is the same with any existing drawing, show a message dialog to remind the user that this ID already exists.

iv. If the new ID is unique from any existing drawing, save this new ID to a variable `newID`, which will be used by the existing codes to add a new drawing object to the list.

Once you finished all codes, please check to see whether the button Add works.

- f. The function `updateDrawing()` is triggered by the button Update. It will use the drawing object that is currently selected in the dropdown list, and update all attributes (except `DrawingID`, which will not be changed by this function) using the data from the five `<input>` tags, **and then save the object back to the `buildings` array**. This process is the opposite of the Task 1.d, which uses the selected drawing object to fill-in the `<input>` tags. The codes for getting the reference of the currently selected drawing object is already written. (1 pt)

Once you finished all codes for this task, please check to see whether the button Update works. Try to change some attributes value in one drawing and click Update. Then, choose a different drawing and come back to the one that you have changed to see if the attributes are showing updated values.

- g. The function `deleteDrawing()` is triggered by the button Delete. It will use the drawing object that is currently selected in the dropdown list, and delete it from the array `drawings`. Please complete the codes to delete the currently selected object. (1 pt)

**Hint:** Please refer to the function `drawingInfoListChange()` and `updateDrawing()` for the approach to get the currently selected index of the drawing object.