Lab 6 – Comp 140

Due: Week of November 10-14 at the start of your lab

Purpose

1. Demonstrate use of client-side JavaScript within an HTML form to create dynamic and responsive feedback by defining event handlers.

Overview

Read this section for the summary of the lab.

1. Building on the HTML form work completed from the previous lab you will enhance its responsiveness and the functionality with JavaScript event handlers. An event handler does nothing most of the time while the HTML page is displayed – its only purpose is to activate when a specified event occurs such as a mouse click in a form element or a button click on a button. When the event handler detects an event has occurred, the handler tells the browser to perform some task defined in some JavaScript (in this lab's case, a JavaScript function).

   The event handler process works like this:

   - Any element in the HTML page you want the user to be able to interact with such as a button, a text box, or a drop down selection can detect appropriate events like 'click', 'mouse over', 'mouse down', 'blur' and so on.

     Events example with a button element:

Event name	**Description**
Click	When you click on the button
Mouse Over	When you move your cursor over the button
Mouse Down	Use the mouse to click down but before you release it
Focus	When you use the tab on the keyboard to tab to the button
Blur	When you tab away from the button

   - Using JavaScript you can provide instructions for the browser when an element's event occurs (for example, the user clicks on the submit button). These instructions are what is called an event handler. If you don't provide any instructions or event handlers for an event, then should that event occur, the browser simply will ignore it.

   - You will add appropriate event handlers for each of the form elements (radio buttons for selecting the pizza type, selection list for pizza size,

checkboxes for selecting optional toppings, and so on). The radio buttons and checkbox buttons need 'click' event handlers while the selection list needs a 'change' event handler.

- When the appropriate event is activated for an element (for example, 'click' for the submit button), the browser will run the instructions defined for that event handler, if there is one.

2. JavaScript supports a number of ways to define event handlers for elements. An early method involved using inline attributes like this:

```
<input type= "radio" name= "pizza"
value= "cheese" onclick= "PizzaType();">
```

Here the `onclick` attribute means that when the user clicks on this element, run the event handler named `PizzaType` (which will be defined as a JavaScript function somewhere in the HTML file or JavaScript file).

This method will work but it is not considered good form because you are inserting JavaScript function calls all throughout your HTML. This makes it harder to read and figure out JavaScript problems.

A recommended approach is to define within one JavaScript file all the form elements' event handlers and functions when the HTML page loads into the browser. This contains the JavaScript within a single file rather than scattered among the various HTML elements.

JavaScript allows you to create a listener for HTML elements. The listener's job is to do nothing until it 'hears' its event. For example, the radio buttons for the pizza type will each have an 'event listener' waiting for a 'click event'. When activated, the event listener will pass control temporarily to its event handler to complete those instructions.

Here is an example of a JavaScript instruction to add a new event listener to an element (named `element`). The first parameter is "`click`" and it means listen for a `click` event for `element`. The second parameter is `costPizzaType` and it is the name of JavaScript function defined somewhere in the script file. This is the event handler. The third parameter is **false** and is always **false** except under special conditions.

```
element.addEventListener( "click", costPizzaType, false );
```

3. Once we have defined the appropriate listen events to each of the form elements, we can go ahead and write the event handlers as JavaScript functions (e.g. `PizzaTypeUpdate`).

4. The pizza order form will display updated cost amounts as the user makes selections from the HTML order form.

Resources

- Textbook – chapter on JavaScript
- Online Core JavaScript guide
- JavaScript coding convention and from the Mozilla Developer Center
- Table of browser issues with some events
  http://www.quirksmode.org/dom/events/

Preparation

Read this section for a review of the information this lab is presenting to you. Information you need is available in Section 4 – Javascript on D2L.

1. JavaScript – what it is used for
2. JavaScript – language syntax rules
3. JavaScript and DOM

Description

Read this section for a description of what you will be doing in this lab.

1. Modify pizza order form from the previous lab so that as the user makes a selection for a pizza size, type and toppings, an updated total cost is shown on the web page.

   For example, when the user selects a type of pizza, the price next to the selected pizza and a running total is shown on the form near the top of the page where it is visible. If the user decides to change a selection, the total is updated. Figure 1 shows a sample of the order form's lower part.

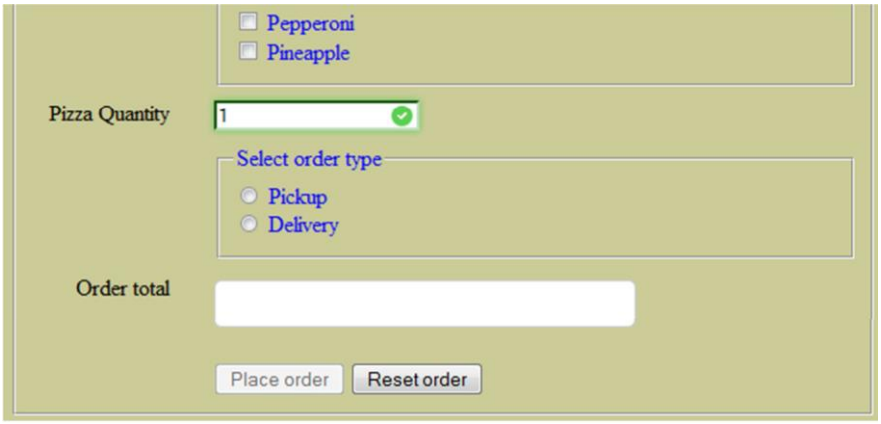

Figure 1 Pizza order form

2. The order form's submit button (the one having the text 'Click here to place order') must be disabled by default until two of the critical pizza options have been selected – i.e. both Pizza Size and Pizza Type selections have been made. Also, clicking the reset button should disable the submit button.

3. Use the prices as shown from lab 2. The default selection for the pizza size should be 'individual'. The default selection for the pizza type should be 'cheese'. The default selection for quantity should be 1. There are no defaults for any toppings.

4. Use JavaScript to show the current date and time near the top of the order form display.

5. In an HTML document any JavaScript program code is contained within the tags

```
<script  type="text/javascript">

  … JavaScript functions and variables must be located inside here

</script>
```

just as all CSS are enclosed by the tags

```
<style  type="text/css">

  … any embedded CSS must be defined inside here

</style>
```

Normally all JavaScript and CSS is kept separate within the HTML document's head section. It doesn't matter which is defined before the other (sometimes CSS is defined before JavaScript). Occasionally, these may appear within the HTML body section as well.

The JavaScript you need to complete in this lab will be within one script file named `order.js` kept in the `script` folder. For example: (don't write them yet)

```
function  costPizzaType()  {

}

function  costPizzaSize()  {

}

function  costPizzaTopping()  {
```

```
}
```

…and other JavaScript functions…

Process

Follow these steps to complete the lab work. Take your time doing this lab especially parts 1 to 6 —if you rush through steps without reading instructions carefully, you may miss details. If you "copy and paste" this lab's code into your HTML code, you will have to rewrite any double quote characters! Supplemental material which explains key JavaScript and DOM concepts presented in this lab are available in Section 4 – Javascript on D2L

1. Open the File Explorer and create the following new folders on your H: drive:

   ```
   comp140\lab06
   comp140\lab06\PizzaPalace
   comp140\lab06\PizzaPalace\images
   comp140\lab06\PizzaPalace\script
   ```

   On the H: drive copy all your previous lab 5 work (the html files, php script and the image files) into your new Lab 6 folders.

2. Open WinSCP and copy the new lab06 folder to your `deepblue` account inside the `public_html\comp140 folder.`

3. Confirm all your navigation links in the Lab 6 folder work correctly in the browser with the absolute URL `http://deepblue.cs.camosun.bc.ca/~cstxxx/comp140/lab06/PizzaPalace.`

4. Edit the `order.html` file.

   At the bottom of the file just before the `</body>` tag, add in this HTML element:

   ```
   <script language="javascript" type="text/javascript"
   src="script/order.js"></script>
   ```

   You are adding this JavaScript file at the end of the body element so that the page is fully rendered before the JavaScript can add in the appropriate event handlers.

5. Download the starting JavaScript code located below this lab assignment in D2L and change the `order-script.txt` file format extension and name to the `order.js`. Place it in the newly created `script/` folder.

This JavaScript file initially sets up all the required event listeners for you. You will have to provide the JavaScript code for the event handlers in this lab so keep this `order.js` file open in DreamWeaver or any other text editor. The *skeleton* JavaScript functions that are the main focus of this lab are defined in the file for you. You will be adding JavaScript code to them in this lab to make the work.

6. Edit the `order.html` file. Scroll to the HTML form where you have the radio buttons defined for the pizza type and make sure you have defined name, id, and value attributes in the `<input>` tags as follows:

```
<input      type     = "radio"
            name     = "pizza"
            id       = "cheese"
            value    = "cheese"> Cheese

<input      type     = "radio"
            name     = "pizza"
            id       = "pepperoni"
            value    = "pepperoni"> Pepperoni

<input      type     = "radio"
            name     = "pizza"
            id       = "hawaiian"
            value    = "hawaiian"> Hawaiian
```

7. Edit the `order.js` file. Edit the JavaScript function named `costPizzaType` as shown. (Do not define a new function with that name.) This JavaScript function performs two tasks:

   a. Determine which radio button was clicked out of the three different pizza types: cheese, pepperoni and Hawaiian.

   b. Whichever radio button was selected, the value attribute of that button will be used to determine the cost of that pizza's type.

      For example, if the cheese pizza is selected, then the value of that radio button will be "cheese" and the JavaScript function will set the pizza type cost to 5. Initially you will construct the function with a simple JavaScript `alert` statement to confirm that the `onclick` event handler is working correctly.

   c. JavaScript variables named `strPizzaType` and `numTypeCost` hold the name of the selected pizza type (one of either 'cheese', 'pepperoni', or 'hawaiian'), and the cost amount for that selected type. The 'str' prefix for `strPizzaType` is to remind you that this is text (string) information.

```
1    ...just below where the comment states Lab 6 work starts
```

```
2       from this point: (but not within the comment body).
3
4     //  Start of the JavaScript code section.
5
6     //  Global variables are defined here.
7
8       var strPizzaType = ""; // name of selected pizza type
9       var numTypeCost = 0;    // cost of selected pizza type
10
11    //  JavaScript functions are defined here.
12
13      function costPizzaType() {
14
15          // version alpha
16
17          alert("Selected pizza type cost is " + numTypeCost);
18
19          return numTypeCost;
20
21      } // end function costPizzaType
```

8. Save your work and press F12 to preview it in a browser (preferably Google Chrome browser as it currently supports all the HTML 5 features).

   If the Microsoft Explorer browser starts, it may detect that your HTML is attempting to run JavaScript and may warn you about possible security issues. Click on the warning bar and select the "Allow Blocked Content…" option. A Security Warning popup will appear. Click Yes to let this file run active content.

Figure 2 Microsoft Internet Explorer detecting JavaScript in HTML

Click on any of the pizza type selections: cheese, pepperoni, or Hawaiian and the `click` event handler is triggered calling the JavaScript function `costPizzaType`, which determines which of the radio buttons was clicked and shows the selected type's cost (currently zero) in a JavaScript alert box.

If you did not get expected results at any point, confirm the following:

a) The `click` calls the exact same name as the JavaScript function name – case matters!
b) Confirm that the `name` attribute for your radio buttons has value "pizza"
c) Check usage of the double quotes in the JavaScript function.
d) Check that each of the value attributes is checked in the JavaScript function in the "if" statement.

e) Check that the if comparisons use double equal signs as ==
f) Check the each JavaScript statement ends with a semicolon
g) Check that braces are matched { } in the JavaScript

Use the Error console and Firebug tools in Firefox. Or check the common JavaScript errors shown at the end of this lab write up.

9. Modify the JavaScript function `costPizzaType` to add in new JavaScript code as follows. Remove the previous "alpha" version of the JavaScript code in that function. Do not add a second function with the same name. The text on lines 13 and 38 are shown 'grayed out' to indicate that you don't need to re-enter them.

```
13      function costPizzaType() {
14
15          // version beta
16          /* Determine which of the three radio buttons was
17             clicked by user by finding the one having the
18             checked property.
19           */
20           var radioElements =
21              document.getElementsByName("pizza");
22           var radioValue = 0;
23
24           for (var i=0; i < radioElements.length; i++ ) {
25                if (radioElements[i].checked) {
26                    /* Found the radio button clicked,
27                       return its value.
28                    */
29                    radioValue = radioElements[i].value;
30                    break;    /* end the loop */
31                } // end if
32           } // end for
```

```
33
34          strPizzaType = radioValue;
35
36          alert("You chose the " + strPizzaType + " pizza.");
37
38          return numTypeCost;
39
40      } // end function costPizzaType
```

Save your work and press F12 in Dreamweaver to test it in a browser.

10. Modify the JavaScript function `costPizzaType` to add in new JavaScript code as follows. The light gray text is existing JavaScript code.

```
13      function costPizzaType() {
14
15          // version prod
16          /* Determine which of the three radio buttons was
17             clicked by user by finding the one having the
18             Checked property.
19           */
20           var radioElements =
21              document.getElementsByName("pizza");
22           var radioValue = 0;
23
24           for (var i=0; i < radioElements.length; i++ ) {
25               if (radioElements[i].checked) {
26                   /* Found the radio button clicked,
27                      return its value.
28                   */
29                   radioValue = radioElements[i].value;
30                   break;   /* end the loop */
31               } // end if
32           } // end for
33
34           strPizzaType = radioValue;
35
36           if (strPizzaType == "cheese") {
37               numTypeCost = 5;
38           } else if (strPizzaType == "pepperoni") {
39               numTypeCost = 7;
40           } else if (strPizzaType == "hawaiian") {
41               numTypeCost = 9;
42           } else {
43               numTypeCost = 0;
```

```
44          }
45
46          alert("You chose the " + strPizzaType + " pizza " +
47                "  and the cost is " + numTypeCost);
48
49          return numTypeCost;
50
51      } // end function costPizzaType
```

11. Add new lines in the JavaScript which declare two new variables `strPizzaSize` and `numSizeCost`.

```
...just below where the comment states Lab 6 work starts
   from this point: (but not within the comment body).

//  Start of the JavaScript code section.

// Global variables are defined here:

   var strPizzaType = "";   // name of the selected pizza type
   var numTypeCost = 0;     // cost of the selected pizza type
   var strPizzaSize = "";   // name of the selected pizza size
   var numSizeCost = 0;     // cost of the selected pizza size

   function costPizzaType() {
     ...
         return numTypeCost;

    } // end function costPizzaType

    function costPizzaSize() {

    } // end function costPizzaSize
```

12. Add in the new JavaScript function `costPizzaSize()` after the function `costPizzaType()`. This second JavaScript function determines which of the three select options was made (individual, medium, or large). This event handler function is easier to code than the one for radio buttons since we only need to find which option was made. Confirm you have defined the `select` as follows with the name attribute:

```
<select name="size"

        Id="size">
```

13. Modify the pizza size option elements' value attributes as follows. The default size is 'Select…" with value of zero.

```
<option value= "0" selected="selected"> Select… </option>

<option value= "1"> Individual </option>

<option value= "2"> Medium </option>

<option value= "3"> Large </option>
```

14. Enter the following JavaScript function for `costPizzaSize`. The JavaScript variable `elt` is used to hold the select element.

```
53    function costPizzaSize() {
54      /* Retrieve the pizza size from selection */
55
56      var elt = document.getElementById("size");
57
58      numSizeCost = parseInt( elt.value );
59
60      strPizzaSize = elt.options[elt.selectedIndex].text;
61
62      alert("size is "+ strPizzaSize + ", cost is "
63                        + numSizeCost );
64      return numSizeCost;
65
66    } // end function costPizzaSize
67
68    //  End of the JavaScript code section.
69
```

15. Save your HTML file and preview it in the browser. Confirm that the selections for pizza type and pizza size appear in an `alert` popup correctly.

16. For this lab we will abstract the cost of each topping as $1 each. Since the toppings are optional combinations, this cost can range from zero (no toppings) to five (all toppings selected). Make sure you have defined the name attribute as follows:

```
<input      type ="checkbox"
            name ="topping[]"
            value ="tomato" >
```

17. Review the JavaScript code and note how there are two separate JavaScript functions – one function named `costPizzaType` and one function named `costPizzaSize`. Note how JavaScript functions are declared with the "`function`" keyword followed

by parenthesis, followed by an open curly brace {, followed by one or more lines of JavaScript program code, and then lastly an ending curly brace }.

Note that JavaScript statements end with a semicolon **;**.

JavaScript is 

18. Add in the global variable declaration for `numToppingCost`.

```
...just below where the comment states Lab 6 work starts
   from this point: (but not within the comment body).

//  Start of the JavaScript code section.

// Global variables are defined here:

   var strPizzaType = "";   // name of the selected pizza type
   var numTypeCost = 0;     // cost of the selected pizza type
   var strPizzaSize = "";   // name of the selected pizza size
   var numSizeCost = 0;     // cost of the selected pizza size
   var numToppingCost = 0;  // cost of the selected toppings

   function costPizzaType() {
```

19. The `costTopping` JavaScript function is similar to the first two functions except that it needs to count the number of checkboxes selected rather than determine which one radio button was selected. Each checkbox found to have been "checked" will increment the checkbox counter.

```
function costTopping() {

 /* Determine which checkboxes were clicked -
     The one(s) having the checked property set.
 */

 var checkboxElements =
          document.getElementsByName("topping[]");

 var checkboxValue = 0;   /* set count to 0 */
 for (var i=0; i < checkboxElements.length; i++) {
    if (checkboxElements[i].checked) {
         /* Found a checkbox that was checked,
             increment count.
         */
        checkboxValue += 1;   /* increment count */
     } // end if
```

```
85      } // end for
86
87      numToppingCost = checkboxValue;
88
89      alert("toppings cost is " + numToppingCost);
90
91      return numToppingCost;
92
93    } // end function costTopping
```

20. Save your work and confirm the selection of toppings is working correctly in the browser preview.

21. The pizza quantity selection is an input form element. This event handler function is easier to code than the ones for radio buttons and checkboxes. Confirm the *name* and id attributes are correct:

```
<input type= "number" id= "quantity"
       name= "quantity" value= "1" min= "1" max= "20"
       required>
```

22. In the JavaScript code section add the declaration for the new global variable:

```
var numQuantity = 1;   /* number of pizzas selected */
```

after the line `numToppingCost = 0;` declaration.

After the `costTopping` function the JavaScript function `getQuantity` is defined as:

```
95    function getQuantity() {
96      /* Retrieve the number of pizzas from selection */
97
98      numQuantity = parseInt(
99            document.getElementById("quantity").value
110       );
111
112     alert("quantity is "+ numQuantity );
113
114     return numQuantity;
115
116   } // end function getQuantity
117
```

23. Save your work and confirm the selection of pizza quantity is working correctly in the browser preview.

24. The four components of the total cost (pizza type, size, toppings and quantity) are stored as JavaScript variables named `numTypeCost`, `numSizeCost`, `numToppingCost` and `numQuantity` respectively. We need a new JavaScript function which will use these values to determine the total cost of the order. On the line following the definition of the `getQuantity` function, create the function `calcTotal` as

```
118   function calcTotal() {
119
120     /* Determine the cost of the completed order. */
121
122     var numOrderTotal =
123        (numTypeCost + numSizeCost
124       + numToppingCost) * numQuantity;
125     alert("Total cost is " + numOrderTotal );
126
127     return numOrderTotal;
128
129   } // end function calcTotal
130
```

In each of the JavaScript functions `costPizzaType`, `costPizzaSize`, `costTopping` and `getQuantity` add in the new JavaScript line

```
calcTotal();
```

just after the line that calls the JavaScript function `alert()`.

25. Save your work and confirm that the selections of pizza type, pizza size, toppings and quantity cause the total to appear in the JavaScript alert window correctly. If you are not getting results, check the error console in the Firefox browser (assuming you are using Firefox). Ensure all the matching braces and parentheses are present. *Make sure* each statement of JavaScript ends with a semicolon.

26. The next step is to add a new row to the form to display the calculated total amount as the user makes pizza order selections. Add in the following HTML just after the `</div>` for the pickup/delivery selection.

```
<div class= "row">
  <span class= "form_label"> Order total </span>
  <span class= "form_element">
        <input type  = "text"
               id    = "total"
```

```
                    value = "0"
                    readonly = "readonly"
                    size = "4" >
    </span>
</div> <!-- end row -->
```

27. Modify the JavaScript function `calcTotal` to assign its calculated value of `numOrderTotal` to this new form element named `total`. If you have completed the lab up to this point you should be able to figure this out (hint: the form element for `total` has an `id` of "total", second hint: the line starts with document.getElementById…).

28. Comment out the functions' calls to the `alert` function so the alerts no longer display.

29. Save your work and confirm in the browser preview that all the order selections cause the correct total amount to appear in the `total` textbox at the bottom.

    The total amount can be formatted to appear as dollars and cents using the built-in JavaScript `Math` function `toFixed`.

    Change the line in your `calcTotal` JavaScript function that assigns the `numOrderTotal` value to the `total` form element:

    ```
    document.getElementById("total").value ="$" +
    numOrderTotal.toFixed(2);
    ```

30. For the form's reset button when it is clicked, the JavaScript function, `resetForm()`, needs to do five tasks: set the four variables `numTypeCost`, `numSizeCost`, `numToppingCost` and `numQuantity` back to their initial values (zero, and 1 for `numQuantity`) and set the `disabled` property for the submit button back to true. Do not use the `var` keyword when re-initializing the variables – `var` tells JavaScript to create a new variable with that name.

    ```
    document.getElementsByName("submit")[0].disabled=true;
    ```

    Also, the Submit button should be initially disabled by default.

    ```
    <input type     = "submit"
           value    = "Place order"
           name     = "submit"
           disabled = "disabled">
    ```

    Note: If you had used the `<button>` element instead of the `<input>` element for defining the reset button, then you need to edit the `order.js` event listener for the reset button. (line 52)

```
buttons  = document.getElementsByTagName("button");
```

31. The form's `submit` button is enabled once the Pizza Type and Pizza Size form elements are both selected.

    A business rule is a mandatory requirement from the client regarding the implementation of the application. For this lab you will implement the business rule that the submit button is only enabled once both the pizza type and pizza size are selected.

    Add an if statement within the `calcTotal` function to check if both the `numTypeCost` and `numSizeCost` costs are more than zero – if so, then enable the Submit button on the form. Write the appropriate expression for JavaScript for this if statement. JavaScript program code does not use the English words "and", "or", "not" for logical operators, so you may need to look up in your JavaScript notes what the equivalent operators look like (hint: they are two typographic symbols).

```
if (numTypeCost  is  > zero  and numSizeCost  is  > zero)  {

  document.getElementsByName("submit")[0].disabled=false;

       }
```

32. Add the necessary JavaScript to show the current date and time at the top of the page. Check the course's online JavaScript notes for using the JavaScript `Date` object.

    HTML can be dynamically changed through JavaScript and the DOM (Document Object Model).

    After the `<div  id="middlebox">` line and before the `<form>` tag, add the following new `div` element

```
<div  id="today">
      Hello !
</div>
```

    Add the new JavaScript code within the function `startForm`:

```
     function  startForm()
{document.getElementById("today").innerHTML  = new  Date();

 }
```

    Save the HTML file and observe the results in the browser preview. The text Hello! should have been substituted with the current date and time.

33. Use DOM to alter the background colour of the `total` form element when the order total is calculated. The "color code" can be any of the CSS colour codes such as a named colour like "green", or an RGB notation like "`#00ff00`".

```
document.getElementById("total").style.backgroundColor = "color code";
```

34. Use the JavaScript event listener and event handler examples in this lab to create a new event listener and event handler for the pickup/delivery radio buttons. (Hint: review the JavaScript code for the pizza type radio buttons.) Make it so the client obtains a 15% discount on the total order if the pickup option is selected. More hints: make sure the pickup and delivery radio buttons have the same name attribute value (which you can make anything you like); use the JavaScript code in the `window.onload` function in your `order.js` script to see how the event listeners for the pizza type buttons are defined (the first 12 lines) – copy those lines and change the appropriate parts; define a new JavaScript function `getDiscount` to act as the event handler; make the event handler set a new JavaScript variable `numDiscount` to 0.15 if the pickup option was selected; otherwise, set it to zero, then call the `calcTotal` function to update the total value on the form.

Hand In

1. [20 marks] When you want to submit the lab work, send me an email message (julia.hunerli30@online.camosun.ca) with the subject: Comp 140 Lab 6 (where John Smith is your name).

   The body of the message should contain the URL to your work above, e.g. the URL

http://deepblue.cs.camosun.bc.ca/~cstxxx/comp140/lab06/PizzaPalace

   a) state how much of the lab you completed (e.g. completed, or completed up to step 20, etc). You will get part marks for as much as you have successfully completed.

   b) (optional) indicate roughly how much time used working on this lab.