CISC 181 Lab 5 Modifying a JavaScript Program

**This lab requires that you download the file Lab 05.zip from onQ. Don't be dismayed by the length of the lab: The text is mostly explanatory. You will have to pay attention to what you're doing, however.**

Submission guidelines

The problems on this lab require a mix of solution types. Place the following into a zip file called **Lab 05 solutions.zip**:

* Your completed Lab 05.docx file
* Your completed beancounter.html file

Upload that .zip file to onQ by the lab's due date and time.

Programming requires that you type everything to an exact specification, and this is not as easy as it may seem. Professional programmers frequently work in pairs, partly so that one programmer can spot typographical errors or omissions made by the other programmer. Unfortunately, working on this lab by yourself, you do not have the luxury of the extra pair of eyes, but you do have the advantage of being able to copy and paste some code from this document.

As with Lab 2, for this lab you will need a plain text editor; ideally one that knows just a little about the web markup language, HTML (which we will look at in more detail later in the course) and JavaScript program code. My preference, again, is for Visual Studio Code (VS Code). You will not lose points on this lab for using a different editor, but you should not try using Microsoft Word or any program like it that inserts formatting codes into its documents.

JavaScript is a programming language that all modern web browsers understand. As a programming language, it has its strengths and weaknesses, but one of its strengths is its ubiquity; it is everywhere! All but the very simplest websites use it extensively, so it is worth your while to understand a bit about it.

I have written a JavaScript program for this lab and built it into the webpage **beancounter.html** which you can extract to a folder on your computer from Lab 05.zip. Have a look at it by dragging it from a file listing on your computer onto an open web browser. The program will run automatically, waiting for you to place a fake coffee shop order. It should look very much like this:

Table

Description automatically generated

Okay, start your text editor and then open beancounter.html in it. It's quite a long file. On my Windows machine, the top part of the file looks like this in Visual Studio Code:

A screenshot of a computer

Description automatically generated with medium confidence

On a Mac, in a different editor, or even in the same programs but with a different configuration, the file may look a little different, but here are some things to note:

* Good code editors try to make the programmer's job easier by colouring certain kinds of elements differently from others. Thus, for example, all the numbers are a sort of pink (fuchsia, maybe?) in VS Code.
* The file's text begins with the line "<!DOCTYPE html>". The editors use line numbering to identify this as line 1. If the editor you are using does not show line numbers by default, see if there is an option to turn them on. (If there is no such option, there may be an indicator somewhere on your editor's interface showing you the number of the line that is currently being edited.)
* The file has two large sections of note. The first lies between the <script> and </script> tags (lines 6 and 164, respectively) and is where most of the JavaScript code can be found. The second sits lower down inside the <form> and </form> tags (lines 172 and 223), and it provides both the raw elements of the user interface (the button inputs and text box outputs) and some values for many of the JavaScript variables. Think of the form, therefore, as providing the input and output devices for the program found within the <script> and </script> tags. When you load the page into a web browser – whether from a distant server or from your own computer – the browser interprets the form and some style information it retrieves from the server.
* Styling for the user interface – details of colour, how the boxes and buttons are drawn and placed, etc. – are contained in a file on the sites.cs.queensu.ca server that is imported on line 5. We will learn a bit about styling HTML and about HTML itself later in the course.

SAVE YOUR WORK IN YOUR EDITOR FREQUENTLY! Ctrl-S (holding down your Control or Ctrl key and pressing S) will likely do this quickly and easily. If you really mess up, it may be best to start again with a freshly extracted copy of beancounter.html. When you are asked to do any editing, make the changes in your text editor, save your work, then refresh your browser window or drag the revised file to your browser again to see how those changes are reflected in the running program.

1. (1 mark) The program has a keypad to enter quantities, and it runs subtotals (updated using the button marked "Enter") so that the user can, for example, enter complex orders like "2 espressos, 3 short lattes with two espresso shots, 4 tall cappuccinos, and 4 grande Americanos." Use the program to enter this order,
   * click on "2" on the keypad
   * then click on "ESPRESSO" (in the "Drink Type" box)
   * then click on "Enter" (in the "Calculations" box)
   * then click on "3" on the keypad
   * then click on "Short" (in the "Cup Size" box)
   * then click on "2" (in the "Shots (of espresso)" box)
   * then click on "LATTE"
   * then click on "Enter"
   * … and so on…
   * then, finally, click on "Total"

How much, **tax included**, is 2 espressos, 3 short lattes with two shots of espresso, 4 tall cappuccinos, and 4 grande Americanos? (Note: Espresso only comes in one size, so if you place an order for espresso, the "Cup Size" will be ignored.)

$29.02

Go here, <http://sites.cs.queensu.ca/courses/cisc181/Lab_05_check.txt> to check your answer, and if you do not get the same amount, try clicking on the "Clear" button and starting over. It is rather important to the rest of the lab that you understand how to use the program.

1. (1 mark) cupSize is the name (or *identifier*) of a variable in my JavaScript program, and clicking on any of the buttons labeled "Short," "Tall," or "Grande" causes that variable to be loaded with a different value. A JavaScript variable **should be *declared* in a var statement** before it is used. Note the line numbers that your editor puts down the left side of the code in its editing window. Which line is the variable cupSize declared on? (Hint: Look for the "var" keyword in front of the identifier "cupSize.").

16

1. (1 mark) When a variable is declared, it *may* also be *initialized* (given its initial – meaning first – value) using the assignment operator (an equals sign). What is the initial value of cupSize?

0

1. (2 marks) Nationalist sentiments being what they are, Head Office wants us to "Canadianize" our product line, and of course, our Canadian sales taxes are different from the American taxes built into the program currently, so that needs to be changed, too. As you make each change, test it in your browser. **For both marks, your program must run correctly after making these changes.** Here are the changes you are to make:
   1. The drink known as Americano is to become Canadiano. (Note that there are **nine** occurrences of Americano, spelled with upper- and/or lower-case letters, forming parts of identifiers in some cases, in the program and the form. Make sure you change them all, **preserving the case, upper and/or lower, of the characters as you go**.) Your editor likely has a search-and-replace feature that lets you specify whether you want to consider character case in your search. Make sure you select that option. **Getting the character case incorrect may mean the difference between your program working and not working.**
   2. There is a variable called taxRate that is currently being initialized to 0.0848 (which is the tax rate for some American state). This value should be changed to 0.13, representing Ontario's 13% Harmonized Sales Tax (HST). (Note that Ontario charges a reduced tax rate for total restaurant orders of $4.00 or less, but we will not worry about that.)

Having made these changes, save your work, then "Preview" your revised document, and tell me the cost, tax included, of 2 espressos, 3 short lattes with two shots of espresso, 4 tall cappuccinos, and 4 grande Canadianos?

$30.23

Check your answer using a calculator or a spreadsheet program (like Excel) before continuing.

1. This is the trickiest part of the lab. **For full marks, your program must run correctly after making these changes.** Try to preserve the indenting patterns used in the original code.

You are going to introduce a new control on the interface and change the program logic to work with the information it provides. As with all the foregoing, make sure everything is typed exactly as shown, paying special attention to the names of elements like "staffDiscount" with mixed upper- and lowercase letters.

Here is the idea: Head office has decided to offer employees a discount on purchases. It is to be applied to the subtotal amount when the user clicks on the "Total" button and will show up as a separate line item in the "Amounts Owing" box as "Staff Disc". The system user indicates whether a staff discount is to apply to a purchase by means of a checkbox control placed under the "Enter" button. If this control is checked…



… then the discount will be applied when the user clicks on "Total." If it is left unchecked (the default state)…



… then the discount will not be applied.

I have already made some adjustments to the styling of the web page to accommodate the new control and its output field. Here is what you need to do to incorporate them and make them work:

* 1. (1 mark) We'll make changes to the interface first, starting with the checkbox and its label. Find this line down in the <body>…</body> section of the HMTL:

<input type="button" id="enterItem" value="Enter">

Insert these four lines immediately **below** it:

<label for="staffDiscountCheckbox">

<input type="checkbox" id="staffDiscountCheckbox">

Staff discount

</label>

Try to preserve the indenting at that point in the file. Save your work and preview it to see that the checkbox and its "Staff discount" label appear as in the second image above. (You should be able to check and uncheck it at this time, **but it will not affect calculations yet.**)

* 1. (1 mark) Now we will add the output box for the staff discount amount in the "Amounts Owing" area. Find this HTML:

<label for="subtotal">Subtotal</label>

and insert the following lines **below** it:

<input class="number" type="text" id="staffDiscount" value="$0.00" readonly>

<label for="staffDiscount">Staff Disc</label>

Save your work and look at it in your browser to see that the "Staff Disc" label and its field appear under the "Subtotal" field in the "Amounts Owing" section of the interface. At this stage, the interface is complete, **but the new additions still do not affect the operation of the program** because you have not done any actual programming with them. That comes next.

* 1. (1 mark) Head office has said that it wants to experiment with the concept by first giving a 6% discount to staff. As this might change in future, it makes sense for you to introduce a variable that is declared and initialized near the top of the program, perhaps where a great many other variables are declared and initialized. So, find where the variable called price is declared and initialized to 0.00 up in the JavaScript code, and insert a declaration and initialization for a new variable, staffDiscountRate, **below** it as follows:

var staffDiscountRate = 0.06;

Save your work.

* 1. (2 marks) When the user clicks on the "Total" button, the function called total() is executed, causing calculations to happen and numbers to appear in the "Tax" and "Total" fields. This is where we are going to add the logic to test whether the user has "checked" the "Staff discount" checkbox, and, if so, make the amount showing beside "Staff Disc" change from $0.00 to whatever the calculated discount should be, and then to have this amount be used in the "Tax" and "Total" calculations. First, find the code for a function called calculateTotal(), then **replace that code** with this:

function calculateTotal () {

var staffDiscount = 0.00;

if (document.getElementById("staffDiscountCheckbox").checked) {

staffDiscount = Math.round(subtl \* staffDiscountRate \* 100)/100;

}

document.getElementById("staffDiscount").value = "$" + formatPrice(staffDiscount);

var taxes = Math.round((subtl - staffDiscount) \* taxRate \* 100)/100;

document.getElementById("tax").value = "$" + formatPrice(taxes);

document.getElementById("amountOwing").value =

"$" + formatPrice(subtl - staffDiscount + taxes);

}

Having made these changes, save your work, then refresh your browser window and find out the cost, tax included, of 1 espresso, 2 short lattes with three shots of espresso, 3 tall cappuccinos, and 3 grande Canadianos, first **without a staff discount, and then with**?

(without staff discount) (with staff discount)

$22.32

$20.97