

## CSCI 165 (FLEX) Assignment#4

In this assignment, you are asked to create a web page with embedded jQuery code. On the index.html page, a user inputs two integers, then clicks to find out their **sum** or double-click to find out their **difference**.

**Due:** Dec 4, 2021 (Saturday morning) at 8 am

**Submission:** Submit **ONLY** the index.html file to the link " Assignment#4 Desc and Submission"

Note: You don't have to submit the "jquery-3.3.1.js" (the jQuery library), I will use mine when accessing your work.

### Step 1:

Download and open A\_4.zip. Unzip it and open it with Notepad++. The framework of the program is given and you need to complete the program.

### Step 2:

1. When completed, the page should perform the following functions:

- When the page is launched, it has the following user interface.

Replace "Jimmy" with your name and Student ID..

The screenshot shows a web page with a blue header bar containing the text "Welcome to Jimmy's jQuery Demo" in yellow. Below the header, there is a text prompt "Enter two integers (one for each box):". Underneath this prompt are two empty text input boxes. To the right of these boxes is a button labeled "Click or Double Click". Below the input boxes, there is a line of text: "Click the button to find their SUM or Double Click for DIFFERENCE".

- Create embedded style rules (i.e., rules are placed between `<style>...</style>`; DO NOT use inline style) to:
  - format the heading as shown.

- After entering two integers to the text boxes, the user can **click** (once) the button to find out the sum of the two integers. The sum is then displayed.

This screenshot shows the same interface as before, but with the first input box containing the number "22" and the second containing "9". The button "Click or Double Click" is still present. Below the input boxes, the text "Their sum is 31." is displayed in a red color.

- Create embedded style rules (Note: DO NOT use inline style) to:
  - format the result with a color font (use a color other than black).

If the user "click" (once) the button, the sum of the two input integers is displayed

- After entering two integers to the text boxes, the user can **Double click** the button to find out the difference of the two integers. The difference is then displayed.

This screenshot shows the same interface, with the input boxes containing "22" and "9". The button "Click or Double Click" is present. Below the input boxes, the text "Their difference is 13." is displayed in a red color.

If the user "double click" the button, the difference of the two input integers is displayed

- In the case when the first integer is smaller than the second one, the result is still a positive number

**Welcome to Jimmy's jQuery Demo**

Enter two integers (one for each box):

Click the button to find their SUM or Double Click for DIFFERENCE

9 22 Click or Double Click

Their difference is 13.

If the user “double click” the button, the difference of the two input integers is displayed

Another scenario when the first number is smaller than the second, the result is still positive

### How to perform Sum and Difference in jQuery:

**Sum** is performed by the operation “+”. For example, I can assign the result of (**num1** is added to **num2**) to the variable named **sum**:

```
var sum = num1 + num2;
```

**Difference** is performed by the operation “-”. For example, I can assign the result of (**num1** subtracts **num2**) to the variable named **difference**. Since it is the “difference” between the two numbers, the result should always be positive. Therefore, the Math.abs() function is needed to cast the result:

```
var difference = Math.abs(num1 - num2);
```

Note: A number entering to the input box is a string, you need to use parseInt to cast (i.e., convert) it before assigning it to the variable **number** as in the following example.

```
var number = parseInt($("#input").val());
```

### Assumption:

- To simplify the implementation, you can assume the user always enters integers. That is, you don’t have to check whether an input is a non-numeric input or not.

### Assessment

The following penalties will be applied when assessing your work. Penalty will also be applied to overall quality of work.

*(Note: assessment on quality is subject to the decision of the instructor)*

Web page/site do not display anything on the browser (i.e., there are error in your code, fix them before submitting you work)	-100% The most common mistake is “spelling error”. Since the source code of the program is given, a request of sending your work to instructor before deadline and checking your code will not be accepted. Please use the features provided by the IDE, go through the program line by line, word by word and you can fix the error.
Requirements not fulfilled	Up to -100%; it depends on the % of requirements completed
Incorrect file name(s)	-5% per incorrect name (e.g., index.html.html, index(1).html, etc. are incorrect)
Unorganized code, improper code formatting,	Up to -30% (use Tab key instead of “a number of spaces” to make indentation, i.e., when you need to move a code statement to the right with indentation)