Week 4 Tasks for Submission

This document contains a number of tasks for you to attempt.

There are three types of task:

- Tasks labelled as "no submission required, not part of portfolio"
  - They are for you to attempt and practice.
  - They provided assistance for you to develop solutions to other tasks that are part of your portfolio
- Tasks labelled as "PASS Submission Task"
  - These tasks are part of your portfolio geared towards all students
  - The solutions to these tasks must be uploaded for marking by your tutor
- Tasks labelled as "CREDIT Submission Task"
  - These tasks are part of your portfolio geared mainly towards students aiming for more than a pass grade. Of course students aiming for a pass grade may attempt and submit these tasks.
  - The solutions to these tasks must be uploaded for marking by your tutor

Screen Capture Process

You will be asked to take screen captures of your HTML pages, JavaScript code etc.

Windows comes with an application called Snipping Tool which will perform this task.

Mac OSX has some built in keyboard shortcuts (Command-Shift-4). The image is saved on the desktop. https://www.itg.ias.edu/content/keyboard-shortcuts-capture-screen-shot-mac-os-x.

There are also products such as Jing (Win and OSX).

Try to avoid taking screen shots of the entire screen if you are only interested in part of that screen.

Submission Process

Download the files **W03P.DOCX** and **W03C.DOCX** from Canvas.

Paste the required screen captures from the tasks below into the appropriate places within these files.

When complete, use the **File / Export** menu option to generate the files **W03P.PDF** and **W03C.PDF**

Finally log into **Doubtfire** and **submit** both files into the appropriate weekly tasks.

File Locations:

In the following tasks you will often be asked to save a file to a **storage location**. You may use any storage location that you wish. You will not be asked to submit your .html or .js

Template

From the previous week we are using **template_upd.html** for creating Web pages. You modified this file to contain your name and ID in the footer area. Since from this week we are using textboxes for input, you may want to add <form> tags and <button> tags to the <article> area, e.g.

```
<article>
<form>

  <button type="button" id=""> </button>
</form>
<p></p>

</article>
```

JavaScript Tasks

Task 1. (no submission required, not part of portfolio)

Create a HTML file named **w4a.html** based on **template_upd.html**

- Change the title to Testing input and output, description "getElementById and innerHTML" and update other meta tags as applicable.
- Add a script element which specified a file named w4a.js within the **src** attribute.
- Inside the <form> section place the following:
  <label>Your name <input type="text" id="firstname" /></label>
  <button type="button" id="process"> Submit </button>
- After </form>, place the following
  ** **

Create a JavaScript file named **w4a.js** based on **template.js**

Within init():

- Create a local variable **btn** and assign document.getElementById("process") to this variable.
- Assign call to function processName to the onclick event of the btn variable, i.e. btn.onclick=…;

Create a function processName(). This function does not need any parameters.
Within processName():

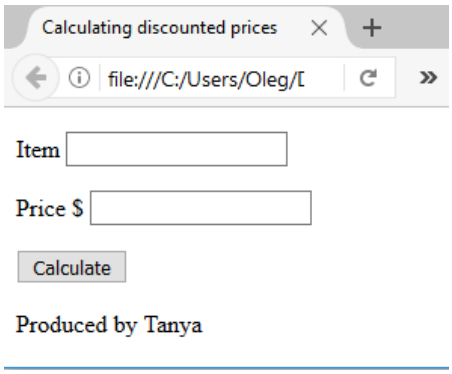
- Create a local variable **first** and assign the value entered into the textbox to this variable: document.getElementById("firstname").value
- Create a local variable **output** and assign document.getElementById("msg") to this variable.
- Use innerHTML to display a string stating "Hello, <first>" – replace <first> with the value entered by the user.

Test w4a.html

Task 2. (no submission required, not part of portfolio)

Create a HTML file named **w4b.html** based on **template_upd.html**

- Change the title to Calculating discounted prices, description "discounting prices based on the old price and discount rate of 20%" and update other meta tags as applicable.
- Add a script element which specified a file named w4b.js within the **src** attribute.
- Inside the <form> section: use <label> and <input /> to create a webpage similar to the one on the screenshot
  First textbox id="item",
  second textbox id="price",
  button id="process"



- After </form>, place the following
  ** New price for the **
  **$ **

Create a JavaScript file named **w4b.js** based on **template.js**

We need an application that will calculated discounted prices based on the old price. Discount rate is a flat 20%.

First create a function **calcDiscountedPrice()** which takes one parameter price and returns the discounted price. Hint: since the discount is 20%, the discounted price is 80% of the current price.

Within init():

- Create a local variable **btn** and assign document.getElementById("process") to this variable.
- Assign call to the function processPrice to the onclick event of the btn variable.

Create a function processPrice(). This function does not need any parameters.

Within processPrice():

- Create a local variable **itemName** and assign the value entered into the first textbox to this variable.
- Create a local variable **oldPrice** and assign the value entered into the second textbox to this variable.
- Since oldPrice is a string, convert it to a number.
- Create a local variable **newPrice** and assign call to the function calcDiscountedPrice() to this variable; don't forget to pass the correct argument to this function.
- Create the output replacing tags with the relevant values. For example, if the user entered item Pillowcase and price as $10 the output should read "**New price for the Pillowcase is $8.00"**.
  Note: your new price should be formatted to 2 decimal places.

Test w4b.html

Task 3. (no submission required, not part of portfolio)

Create a HTML file named **w4c.html** based on **template_upd.html**

- Change the title to Grade from mark, description "Determine subject grade based on the final mark" and update other meta tags as applicable.
- Add a script element which specified a file named w4c.js within the **src** attribute.
- Inside the <form> section:
  use <label> and <input /> to create a webpage allowing the user to enter student ID and

student mark;
use <button> to create a button with id="send".

- After the closing form tag, the output paragraphs should state :
  Student ID _____
  Subject mark _____
  Subject grade _____

The rule is specified in the table below (assume that a mark is an integer, no decimal part allowed):

Mark	**Grade**
Below 50	F (Fail)
From 50 and under 60	P (Pass)
From 60 and under 70	C (Credit)
From 70 and under 80	D (Distinction)
Between 80 and 100 inclusive	HD

Create a **JavaScript** file named **w4c.js** based on **template.js.**

First you need to create a function **markToGrade()** which accepts one parameter mark and returns corresponding grade, except when the mark is not between 0 and 100 returned value is X.
Create a local variable grade.

You will need a set of nested if statements. Although there is more than one way of writing nested if statements here, let's start with validating mark. First check whether the mark is below 0 or above 100 and assign X to grade. Else means that the mark is valid (i.e. it is between 0 and 100 inclusive). So you can start checking if the mark is below 50 and assign the grade accordingly.

```
if (… - condition for invalid mark)
        grade = "X";
else //mark is between 0 and 100
        if (mark < 50)
                grade = "F";
        else //means mark >=50 so need to check the mark is <60
                if (mark < …
```

Note: tutors will not accept badly structured nested if statements, where the CPU has to test conditions.

Within init():

- Create a local variable btnSend and store the reference to the HTML button with id "send".
- Assign call to the function processMark to the onclick event of the btnSend variable.

Create a function processMark(). This function does not need any parameters.

Within processMark():

- Create a local variable **studentID** and assign the value entered into the first textbox to this variable.
- Create a local variable **subjMark** and assign the value entered into the second textbox to this variable.
- Since subjMark is a string, convert it to a number.

- Create a local variable **subjGrade**.
- Call the function **markToGrade()** and assign the returned value to subjGrade.
- Update the first 2 lines of the output on the webpage.

  Student ID <whatever id the user entered>
  Subject mark <whatever value the user entered>

- Use if statement to check whether subjGrade is X. In this case the output on the 3d line should show the error message:

  Subject grade – replace this paragraph with the error message that entered mark is invalid.

  Otherwise the 3d line of the output should be

  Subject grade <whatever is the corresponding grade>

Test w3b.html. Note you have to test every if branch, i.e. marks resulting in every possible grade, including borderline values, such as 0, 50, 60, 70, 80 and 100.

Task 4. (PASS Submission Task)

In this task you will create a basic application that provides a quote on accommodation based on price per night and number of nights.

Create a **HTML** file named **w4P.html** based on **template_upd.html**

- Change the title to "Quote calculator", description "Pass level task" and update other meta tags as applicable including **your student name and ID.**
- Add a script element which specified a file named **w4P.js** within the **src** attribute.
- Inside the article section:
  - add the level 2 heading Accommodation Quote
  - Create a form and between form tags place 2 text boxes with relevant labels to accept Price per night and Number of nights.
    Use <button> to create a button with id="send"
  - Create necessary paragraphs to output standard cost of accommodation, applicable discount and amount due.

Create a **JavaScript** file named **w4P.js** based on **template.js.**

The discount rate is determined using the rule in the table:

Number of nights	**Discount rate**
Less than 5	0%
Between 5 and 10 inclusive	3%
More than 10	5%

Within init():

- Create a local variable btn and store the reference to the HTML button with id "send".
- Assign call to the function processQuote to the onclick event of the btnSend variable.

Create a function named **calcStayCost()** that takes two parameters reflecting price per night and number of nights and returns cost of stay.

From here we proceed in stages.

Stage 1: reading data from the web page.

Within processQuote():

Create local variables to store the following values: price per night, number of nights to be booked, standard cost, discount rate, discount and amount due.

Read the input from the price per night textbox and store it in the relevant variable.

Use alert to display the value from the variable.

Read the input from the number of nights textbox and store it in the relevant variable.

Use alert to display the value from the variable.

Test your program. Do not proceed until all alerts are working properly. If they work, delete alerts.

Stage 2: validating input

Create a function **validateInput()** that takes one parameter and returns true if this parameter is >0 and false otherwise. **Note, this is a generic function that can be used to validate any numeric value.** We will use it to validate first price per night and then we will call it again to validate number of nights.

Within processQuote():

After reading input, convert both inputs to numeric form.

Create a variable *validPrice*, call **validateInput()** passing the variable storing the price per night as an argument to this function and store the returned value in *validPrice*.

Use alert to display the returned value; test both valid and invalid input. If it works correctly, remove the alert() statement.

Use if (!validPrice) to display a relevant error message if the price is invalid (you do not need an *else* branch here)

Repeat the previous step to validate number of nights. Use alert to display the returned value; test both valid and invalid input. If it works correctly, remove the alert() statement.

Use if to check whether number of nights is invalid and if true, display a relevant error message (you do not need an *else* branch here).

Stage 3: calculating standard cost of accommodation.

Within processQuote():

Use if (validPrice AND validQty) and calculate cost of stay by calling the function **calcStayCost()** passing relevant arguments and store the returned value in the variable for standard cost.

Use document.getElementById() to display the value of cost in the relevant paragraph.

Test your program with valid and invalid values. Do not proceed until the program works correctly.

Stage 4: calculating discount.

Create a function **calcDiscount()** that takes two parameters – number of nights and standard cost of stay.

Within this function:

Declare a local variable *dRate*.

Use *nested if* statements to check in what range number of nights is and allocate the corresponding value to dRate. Make sure that this corresponding value is not in %% format but a fraction. **Make sure you nested if structure is not wasting CPU time; check lecture slides, short videos and task 3 for explanation on how to build well-structured nested if.**

Calculate *discount* using *drate* and *standard cost of stay*.

Return *discount*.

Within processQuote():

After you calculated standard cost of stay, declare a variable for storing discount value and call **calcDiscount()** passing the relevant argument.

Create a variable to store amount due and initialise it to standard cost of stay.

Use *if* to check that discount > 0, in this case calculate the amount due after discount.

Display all necessary values on the webpage (i.e. standard cost of stay, applicable discount and amount due.

Test the program using both invalid values, then one valid and one invalid input, make screenshots and paste them into **W04P.DOCX.**

Test the program with both valid values, ensuring you test every branch of the nested **if** including borderline values. **Screen Capture** the **web page** displaying outputs and paste into **W04P.DOCX.**

Copy and Paste the **HTML code** (or screenshot of your code) from your code editor into **W04P.DOCX**
Copy and Paste the **JavaScript code** (or screenshot of your code) from your code editor into **W04P.DOCX**

WordPress Tasks

Task 5. (PASS Submission Task)

Work through the ICT10013_WordPress_intro.docx document (available from Week 4 subject section).

Ensure that your name and ID appear on the Contact page and the dog page. Modify these pages if necessary.

Add a Cat page to your site.
Include 2 different widgets on this page that take your fancy

Add a Bird page to your site.
Include 2 different widgets on this page that take your fancy

Ensure that your menu now includes the Cat and Bird page.

Paste your screen captures of your Dog page and About page into **W04P.docx**