|  |  |
| --- | --- |
| Instrument of Assessment Internally Verified by: | Date: |

|  |  |  |
| --- | --- | --- |
| Assessor: **Gary Nelson** | Date: | |
| IV Sampled by: | Date: | |
|  | Result |  |

**CQA1.1.9: Assessment Material Cover Sheet**

Unit Number: HF4X 34

Unit Titles: Client-Side Scripting for Web Applications

Verification Group: **358**

Outcome(s): **Outcome 2**

Version: **1**

|  |
| --- |
| Student Name: |

**Where the assessment has not taken place under controlled conditions the learner should sign and date to indicate agreement with the statement below.**

“I declare that this submission is entirely my own work, and any content by other authors has been clearly acknowledged and referenced. I understand that disciplinary action will be taken by the college if this is not the case.”

**Student Signature:**  **Date:**

**Assessment Instructions**

* You must submit this assignment no later than ­­­­­­­­­­­­­­­­­­­\_\_\_\_\_ \_\_\_\_
* This is an open book assessment. There are no restrictions on the materials you may use.
* This assessment will be graded Pass or Fail
* To achieve a Pass, you must complete **the task successfully.** This is detailed on the following pages.

If your work is graded as “Fail” you will be given a re-sit. If your work is graded as “Fail” on your second attempt, you may be given a third and final assessment **but only if** **agreed by the members of your staff course team**. For detailed information on the college’s assessment policy please refer to the college intranet policies area or ask your Lecturer.

If you are not satisfied with the grade awarded by your Lecturer, you should speak to your lecturer in the first instance. If you are not satisfied with this response then the Lecturer or Curriculum & Quality Leader will refer you to the college’s appeals procedure.

Impartial advice about the college appeals procedure is available from your studentadvisor.

**Outcome 2 -** Implement programming elements associated with client-side scripting languages.

This assessment takes the form of open-book practical assignments to test your ability to develop, write and implement suitable programming elements within client-side scripting. The tasks collectively cover the requirements outcome 2 and you must demonstrate the use of each at least once in your scripts.

|  |  |
| --- | --- |
|  | Task  Evidence |
| 1. **Variables:** 2. Global variable 3. Local variable 4. Different data types |  |
| 1. **Operators:** 2. Arithmetic operator 3. Comparison operator 4. Logical operator 5. Assignment operator |  |
| 1. **Arrays - use an array: one dimensional or two dimensional or associative** |  |
| 1. **Control Structures:** 2. Selection use an if statement or switch case statement 3. Iteration - use loop |  |
| 1. **Functions:** 2. In-built 3. User-defined 4. Parameter passing |  |
| 1. **Event handling - call an Event** |  |
| 1. **Document Object Model (DOM) manipulation - modify the document using the DOM** |  |

* All code should be commented – describing all logic
* All code should be formatted appropriately
* Display to screen etc… should be taken to mean target an element within the DOM to display results

**NOTE: ALL INPUT FIELDS MUST BE VALIDATED FOR CORRECT FORMATTING & EMPTY INPUTS – AN APPROPRIATE MESSAGE SHOULD BE DISPLAYED BELOW/ABOVE THE INPUT FIELD**

|  |  |
| --- | --- |
| **DOWNLOAD** *assess\_tasks.zip* **FROM MOODLE** | **Evidence covered** |
| **Task 1 – VAT CALCULATOR**  **Open the task 1 file supplied from Moodle**  **Create a small program that allows users to enter an amount of money in pounds.**   1. **When a “Calculate VAT” button is clicked the VAT @22% should be displayed in the VAT field.** 2. **The GRAND TOTAL (***Original + VAT***) should be displayed in the Grand Total field.** 3. **The reset button should clear ALL fields**     **NOTE: The results should be fixed to 2 decimal places** | **In built function**  **Loop**  **Parameter passing**  **Event handling**  **Local variable**  **Event handling**  **Loop**  **User defined function**  **Event handling**  **Comparison operator**  **IF statement**  **or**  **switch statement**  **Event handling**  **Parameter passing**  **Arithmetic operator**  **Event handling**  **logical operator**  **Array**  **Global variable** |
| **Task 2 – Subscribe form**  **Open the task 2 file supplied from Moodle**    **If the user clicks the subscribe button the email and username form fields should be validated before the form submits – If the form passes the validation, it should submit then go to the *subbed.html* file** *(supplied)*   * Email should be validated to a correct email format * Username should allow only numbers and letters   **The user should be able to reset the form at any time by clicking the reset button – looping through the input fields to clear them**  **Task 3 – Nursery Rhyme**  **Open the task 3 file supplied from Moodle**   1. **Create the JavaScript code in the nursery\_script.js file supplied** 2. **Use an IF statement and a comparison operator to match the days in the drop-down menu to the messages in the table below** *(table 1)* 3. **The values should be formatted to upper case after the user selects them, before being matched** 4. **The day the user selects should display ONLY the matching slogan to the page** 5. **The reset button should clear the output from the page**      |  |  | | --- | --- | | *table 1* |  | | **Monday's** | **child is fair of face** | | **Tuesday's** | **child is full of grace** | | **Wednesday's** | **child is full of woe** | | **Thursday's** | **child has far to go** | | **Friday's** | **child is loving and giving** |   **Task 4: Money Program**  **Write a small program to help children learn about money. The idea is when the user types in a number representing £’s, a button can be clicked that will calculate how many 50p, 20p and 10p pieces there are in the amount of £’s entered**     1. **Open task 4 from the task files supplied** 2. **Create the JavaScript code in the money\_script.js file supplied** 3. **The screen shot above is a working example of the finished app**   **NOTE:You MUST demonstrate parameter passing in this task**  **Task 5: Global Generator**  **Open the task 5 file supplied from Moodle**     1. **If green button is clicked a random number between 1 & 26 will be generated and its value sent to appropriate screen message** 2. **Create global variable that increments every time green button is clicked and send the value to appropriate screen message** 3. **The random number generated should be tested:**   **If the number is greater than zero and less than 13 display:**  *“The number generated* ***randNum*** *is less than 13*  **If number is greater than 13 or less than or equal to 26 display:**  *“The number generated* ***randNum*** *is greater than 13”*  **If the number is equal to 13:**  *“The number generated* ***randNum*** *is equal to 13”*   1. **With the letters provided in the tasks downloaded from Moodle create an array with the letters inside and use the *randNum* generated to display a random image in the appropriate message** |

**\*\*\*\*\*\*\*END OF ASSESSMENT\*\*\*\*\*\*\***