Accredited

Computing

GCSE

COMPUTING – J275

Sample Material B1

A452

www.ocr.org.uk/computing

OCR
Oxford Cambridge and RSA

Disclaimer on use of Sample Material

Confidentiality

These tasks are taken from legacy Controlled Assessment tasks, undertaken and submitted by candidates. Where possible, we have removed all identifying information from these assessments. Should any data remain, you are requested to treat this confidentially and inform OCR as soon as possible highlighting the data concerned.

Use of URS Sheets and Sample Material

These tasks have all been moderated as part of the relevant exam series in which they were submitted and the marks submitted have all been allowed to stand. However, schools should bear in mind that this only indicates that the **overall assessment** of the Controlled Assessment is within tolerance and not necessarily each individual mark band. There may be instances where the mark scheme has been applied too generously, or similarly too harshly. This would have been identified in the reports to the centre – but will not be evident from URS alone. The spirit of the release of these samples is to give teachers better understanding of what High, Medium and Low graded coursework would feel like as an entity, rather than exact definitions of requirements for mark bands independently.

The provision of high graded does **not infer** that this is the only, or best way of writing up a Controlled Assessment Task. Candidates are encouraged to map their personal journey through the tasks. Writing frames, or 'guides' for documentation are against the spirit of the coursework and constitute malpractice.

Each set of materials released contains a High, Middle and Low grade band. This should allow teachers to gain good understanding of the general standard of work quality required for each mark band, and as a whole – especially when comparing each set side by side.

Teachers are encouraged to seek further support when they feel clarification is needed in applying the mark scheme. We would also recommend regular CPD in respect of Controlled Assessment delivery and marking.

Accuracy

All work has, where possible, remained unaltered from the original submission. There may well be grammatical errors and poor layout in diagrams. This is to allow better matching of mark band criteria, where specific bullet points refer to quality of Spelling, Punctuation and Grammar, and also ease of navigation etc. Any significant changes are clearly marked. Some data that is perceived sensitive may be blocked out in black.

GCSE Computing Controlled Assessment

Unit A452 Practical investigation

Unit Recording Sheet

Please read the instructions printed on the other side of this form. **One** of these Unit Recording Sheets, suitably completed, should be attached to the assessed work of **each** candidate.

Unit	**A452**			**Year**	
Centre Name				**Centre Number**	
Candidate Name				**Candidate Number**	

	Guidance			Teacher Comment	Mark
Practical investigation	There may be limited evidence of any practical investigation. The evidence supplied is minimal with limited relevance to the set task. The practical evidence may be largely the result of group or teacher led activity with limited input from the student. 0 = no response or responses not worthy of credit **[0 - 5]**	There is evidence of a practical investigation. The evidence supplied is documented clearly and is relevant to the set task. There is evidence of individual research beyond the group activity and any teacher led activity. The practical investigations show signs of planning but there may be omissions made in assessing the consequences. **[6 - 10]**	There is evidence of a well-structured practical investigation. The evidence supplied is well organised and clearly relevant to the set task. There is extensive evidence of individual practical investigation beyond the group activity and any teacher led activity. The practical investigation shows clear signs of planning and a structured approach, providing a complete investigation of the set topic area. The practical investigation has been carried out with skill and due regard to safety issues. **[11 - 15]**	The student has provided and well structured practical investigation and has documented his individual investigation. He has documented the planning that was undertaken and there is evidence of a structured approach.	15 **Max 15**

Effective and efficient use of techniques	The techniques used will produce partially working solutions to a small part of the problem. 0 = no response or responses not worthy of credit **[0 - 3]**	The techniques will be used appropriately giving working solutions to most of the parts of the problem. Some parts of the solution may be executed in a partial or inefficient manner. **[4 - 7]**	The techniques are used appropriately in all cases giving an efficient, working solution for all parts of the problem. **[8 - 10]**	All techniques used are appropriate and all parts of the problem are fully working	10 **Max 10**
Technical understanding	The candidate demonstrates a limited understanding of the technical issues related to the scenario. Little detail is presented. There is limited indication of any evidence provided being analysed. There is limited use of technical terminology. 0 = no response or responses not worthy of credit **[0 - 3]**	The candidate demonstrates a reasonable understanding of the technical issues related to the scenario. The amount of detail presented is adequate to support the arguments. There is some analysis carried out on the evidence collected. Technical terminology is for the most part used appropriately. **[4 - 7]**	The candidate demonstrates a thorough and secure understanding of the technical issues related to the scenario. A wide range of relevant and detailed information is presented. The evidence which has been collected is fully analysed. Technical terminology is used correctly. At the top end of the band, this will be extensive and confidently used. **[8 - 10]**	The candidate demonstrates a thorough and secure understanding of all the technical issues and provides a wide range of detailed and relevant information. Technical terminology has been used correctly.	10 **Max 10**

Conclusions and evaluation	Conclusions are limited with little justification. The solution is presented with limited evidence of testing. Information may be ambiguous or unclear. There is limited reference to evidence. The evidence of written communication is limited with little or no use of specialist terms. There are many errors in spelling, punctuation and grammar. The evaluation may be simplistic with little or no relevance. 0 = no response or responses not worthy of credit **[0 - 3]**	The material has structure and coherence with justifiable conclusions being reached although there may be some omissions. There is evidence that the solutions have been tested for basic functionality. Candidates have produced a sound evaluation which reviews some aspects of the task. There is evidence of good written communication using some specialist terms. There are few errors in spelling, grammar and punctuation. Specialist terms are used appropriately and for the most part correctly. **[4 – 7]**	Thorough and convincing conclusions have been reached, which are borne out by the research carried out by the candidate. The solutions are fully tested and there is little doubt that the solutions presented are fully functional. This material has been presented in a clear and relevant way which is simple to navigate. A high level of written communication is obvious throughout the task and specialist terms/technology with accurate use of spelling is used. Grammar and punctuation are consistently correct. The evaluation is relevant, clear, organised and presented in a structured and coherent format. **[8- 10]**	There is extensive evidence of testing and all the solutions are fully functional. The material has been presented in a clear and relevant way. A high level of communication is obvious and technical terms have been used accurately. Convincing and thorough conclusions have ben reached and student and provided and an excellent evaluation.	9 **Max 10**
				Total/45	**44**

Guidance on Completion of this Form

1 **One** sheet should be used for each candidate.
2 Please ensure that the appropriate boxes at the top of the form are completed.
3 Using the guidance identify the most appropriate mark range for the work and enter the mark awarded for each element in the mark column.
4 Add appropriate comments to assist the moderator in the 'Teacher Comment' column.
5 Add the marks for the strands together to give a total out of 45. Enter this total in the relevant box.

A452 Javascript

1- Describe how this HTML code produces the form displayed in the browser (Fig. 1).

```
56  <h1>Exam Entry Form</h1>
57  <form name="ExamEntry" method="post" action="success.html">
58  <table width="50%" border="0">
59  <tr>
60  <td id="name">Name</td>
61  <td><input type="text" name="name" /></td>
```

- On **line 56**, the <h1> and </h1> tags create the heading of 'Exam Entry Form' displayed by the browser. Therefore the words are displayed in bold and as the tags are <h1> instead of <h2> for example, the largest heading is created.
- One **line** 58 the table's properties are set, with the table having '50%' width. This in effect sets how wide the cells within the table are. Also, as the table border is set to "0%", the border can't be seen therefore there is only space between the two cells on each row.

Just blank space as border has been set to "0%"


```
54  </head>
55  <body>
56  <h1>Exam Entry Form</h1>
57  <form name="ExamEntry" method="post" action="success.html">
58  <table width="50%" border="0">
59  <tr>
60  <td id="name">Name</td>
61  <td><input type="text" name="name" /></td>
62  </tr>
63  <tr>
64  <td id="subject">Subject</td>
65  <td><input type="text" name="subject" /></td>
66  </tr>
67  <tr>
68  <td><input type="submit" name="Submit" value="Submit" onclick="return validateForm(
69  <td><input type="reset" name="Reset" value="Reset" /></td>
```

On **line 59** <tr> creates the first, top, row of the table.

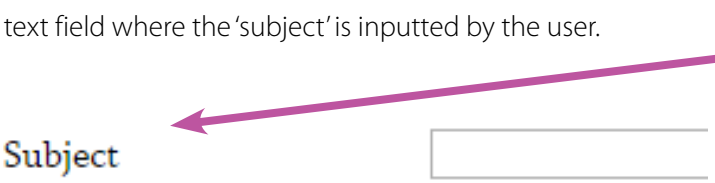
On **line 60** <td> creates the first cell, to the left in the first row, and '**Name**' displays the text:"Name" in that cell.

On **line 61** <td> creates a new cell in this first row of the table, and this is to the right of the first cell. The '**<input type = "text" name ="name" />** creates the text field where the 'name' is inputted by the user.

On **line 63** <tr> adds a second row to the table.

On **line 64** <td> creates the first cell in the second row, and '**Subject**' between '**<td id ="subject">** and **</td>** displays the text: 'Subject' in that cell to the left.

On **line 65** <td> creates the second cell in this second row. The '**<input type = "text" name ="subject" />** creates the text field where the 'subject' is inputted by the user.




```
66    </tr>
67    <tr>
68      <td id="exam">Exam Number</td>
69      <td><input type="text" name="exam" /></td>
```

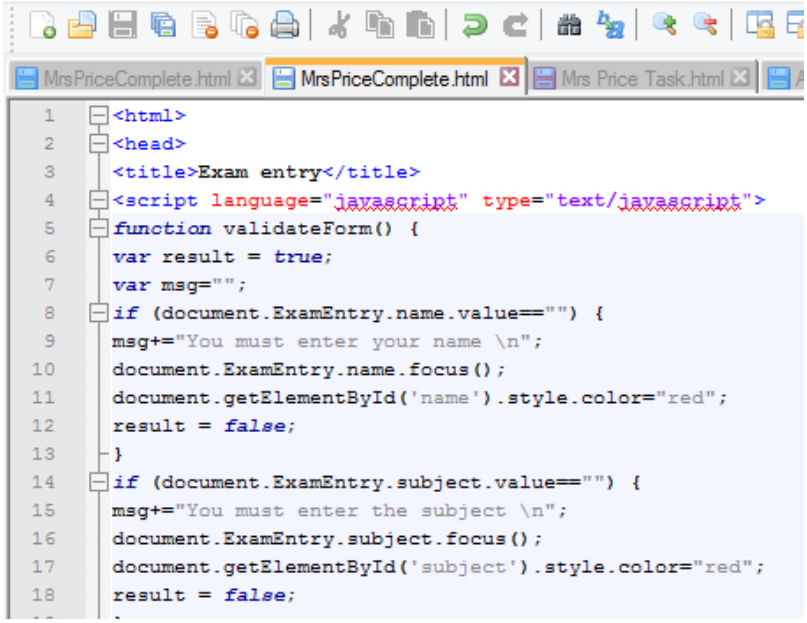
- In **line 67** <tr> adds a third row to the table.
- In **line 68** <td> adds a cell to this row, and **<input type="submit"**… creates a submit button in this cell. The **value="Submit"** means that the text: 'Submit' is displayed inside the submit button.

- In **line 69** <td> adds a second cell to this third row of the table, and in this cell is a 'reset' button, which is created by **<input type="reset"**… . The **value="reset"** means that the text: 'Reset' is displayed in the reset button.

2- Describe how the JavaScript function performs the validation check.

```
78    <td><input type="submit" name="Submit" value="Submit" onclick="return validateForm();" /></td>
```

- Here the onclick event: when the user clicks the 'submit' button the JavaScript function 'validateForm()' is run, so that the form can be validated.



- On **line 6, var result = true**, the Boolean: 'result' is initialised. This means that the form assumes that the user has completed the form without error, so that if there are no errors, then the result (true) can be used to load the success page.

- On **line 7 Var msg=" ";** This is declaring a variable: 'msg'.
- On **line 8** the IF statement: **if (document.ExamEntry.name.value==" ")** detects if the text field has been left blank. A blank text field means the value is nothing, or " ". Therefore an error message can be printed if it is left blank.
- The error message is added to the variable: 'msg' which is basically a variable that stores the different error messages of this form. **msg+="You must enter your name \n";**. adds the error message: 'You must enter your name' to this variable, so that it can be displayed to the user once they press submit.
- On **line 10 document.ExamEntry.name.focus();** puts the cursor on the 'name' text field to alert the user to the error, and encourage them to rectify it immediately.
- On **line 11 .style.color="red";** makes the text: 'Name' red after the error is detected to further draw attention to the user's error.
- On l**ine 12 result = false;** by setting the Boolean: 'result' to false, the code can recognise that the form is invalid and this prevents the success page from being loaded.

```
32  if(msg==""){
33  return result;
34  }
35  {
36  alert(msg);
37  return result;
38  }
39  }
```

On **line 32** the IF statement detects if the variable 'msg' is null.

On **line 33**, if 'msg' is null, therefore there are no errors in the form, and 'result' is returned. Which would load the 'success' web page, since the form is valid. 'Result' here is true, so ***true*** would be returned.

On **line 36, alert(msg)** displays the variable: 'msg' in an alert so the user can read the message. Here, 'result' is false, since 'msg' is not null and **return result;** on **line 37** returns ***false*** to the function

3- Describe how the HTML calls the validation routine.

```
<td><input type="submit" name="Submit" value="Submit" onclick="return validateForm();" /></td>
<td><input type="reset" name="Reset" value="Reset" /></td>
```

- The top line of code here means that as the 'submit' button is clicked it calls the validateForm and therefore the function is executed. If **false** is returned, the submit process does not work; the success page is not loaded

4i- Add another text field to the form to make sure that the user's examination number is exactly 4 digits.

Plan for Code

- I thought that the code for the new text field would be very similar to the code of the name and subject text fields.

I need to:

1. Create a new table row
2. Create a cell for the text showing the user what this text field is for
3. Create a cell for the 'Exam Number' text field
4. Create the 'Exam number' text field

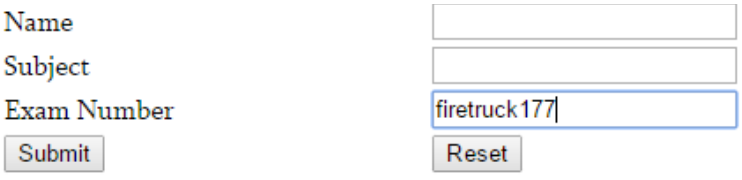
Written Code

```
68  <tr>
69      <td id="exam">Exam Number</td>
70      <td><input type="text" name="exam" /></td>
71  </td>
72  </tr>
```

- Line 68 creates a new table row.
- **Line 69** creates a cell in this new row. The Cell id is "exam" and within this cell is the text: "Exam Number".
- **Line 70** creates another cell in the same row and inside is the text field for the 'Exam Number'. The text field is created by the code: **'<input type="text"** and the name of the text field is "exam".

Testing Code

This shows that the new text field Is displayed, as well as the text next to it to inform the user about what this text field is for.



The user is able to type in text into the text fields.

4ii- Extend the JavaScript code to validate this field to make sure that it is not left blank.

Plan for Code

- Because this is a text field, just like the other two text fields, the validation of it would be almost identical, as the other two text fields are both validated to make sure that they are not left blank.

To validate the 'Exam Number' text field to make sure it is not left blank I need to:

1. Check if the text field had been left blank
2. If the text field is left blank, an error message is added to the list of error messages that is presented in an **alert**. This is to let the user know that they left it blank.
3. Put the focus, cursor, on the text field to further alert the user and draw attention to the error that they made, just like the other text fields are focused when they are left blank.
4. Make the text next to the text field: "Exam Number" which is in the cell to the left of the text field, become red just like the text in the other rows turn red when the other text fields are left blank.
5. Must make var 'result' = false just like the other text fields do so that the error message is actually displayed and the form is validated.

Written Code

```
20  if (document.ExamEntry.exam.value=="")
21  {
22     msg+="You must enter the Exam Number, which must be 4
23     document.ExamEntry.exam.focus();
24     document.getElementById('exam').style.color="red";
25     result = false;
26  }
```

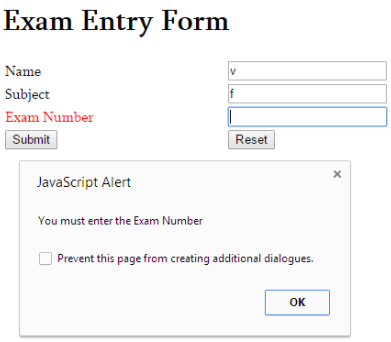
1. **Line 20** contains the IF statement and it checks if the value of the text field is " " i.e. empty space.
2. **Line 22** adds the error message specifically belonging to this error to the var 'msg' so that the error message can be displayed along with any other error messages to alert the user.
3. **Line 23** puts the focus or cursor on the text field.
4. **Line 24** makes the text: 'exam', red to draw attention to the error.
5. **Line 25** makes the Boolean 'result' = false so that the form is not valid and the success page is not loaded. So that the result of the form validation check is false, because there is at least one error, this one.

Testing Code

Red text shows that the field has been left blank.

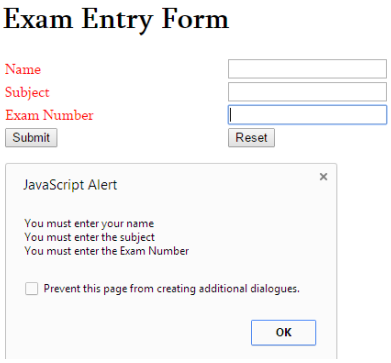
The 'Exam' text field has been left blank but the other text fields have been filled.

JavaScript Alert displays the error message connected to the 'Exam' text field being left blank.



All three text fields have been left blank and therefore three error messages are displayed, showing that the error message linked to the blank 'Exam' text field is added in the var 'msg' and so it is added to the list of error messages.

The success page is not shown, but the user is made to fill in the text field before the success page is displayed.



4iii) Extend the JavaScript code to make sure that the user's examination number is exactly 4 digits.

Plan for Code

To make sure that the exam number inputted is exactly 4 digits long I need to:

1. Detect if the exam number is not a 4 digit number.
2. An error message must be added to the list of error messages contained in the var 'msg'.
3. Put the focus, cursor, on the text field to further alert the user and draw attention to the error that they made, just like the other text fields are focused when they are left blank.
4. Make the text next to the text field: "Exam Number" which is in the cell to the left of the text field, become red just like the text in the other rows turn red when the other text fields are left blank.
5. Must make var 'result' = false just like the other text fields do so that the error message is actually displayed and the form is validated.

- These requirements are exactly the same as the requirements for the previous task (making sure that the text field is not left blank) except number 1, so I devised a plan to link the code for both tasks together so that the code is more efficient.
- Instead of having a whole new 'IF statement' for checking if the exam number is a 4 digit number, it will be added into the condition of the existing IF statement and the error message displayed now in this IF statement is changed so that the error message informs the user on how to enter the correct exam number as well as telling the user that they have not filled this in. This error message then is suitable in the case of either error of the user, whether the text field has been left blank, or if the exam number is not a 4 digit number.

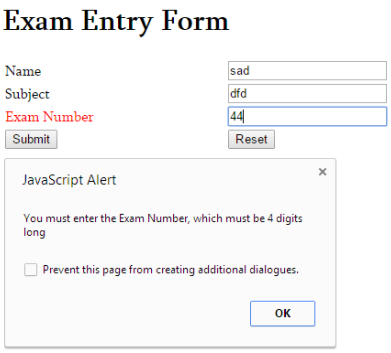
Written Code

```
20  if (isNaN(document.ExamEntry.exam.value) || document.ExamEntry.exam.value.length != 4)
21  {
22      msg+="You must enter the Exam Number, which must be 4 digits long \n";
23      document.ExamEntry.exam.focus();
24      document.getElementById('exam').style.color="red";
25      result = false;
```

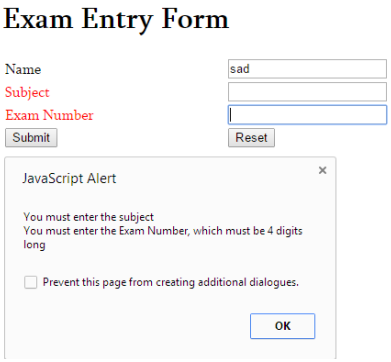
- **Line 20** contains the condition of the IF statement. It actually is the same IF statement of the previous task but the condition has been changed to create a solution to a new task as well as the old one.
- The 'isNaN()' method checks if the exam number is not a number, the '||' sign adds a second condition, which is that if the exam number's length is != 4 (not equal to 4). Together, these conditions check if the user has or hasn't entered a 4 digit exam number.
- **Line 22** contains the new, adapted error message.

Testing Code

The Name and Subject text fields have been filled validly but the exam number has been filled, but not with a 4-digit number. Therefore JavaScript alert is displayed and the text 'Exam Number' is red.



This shows that the error message is added onto the list of errors that are displayed in the JavaScript Alert and that the code still validates the text field so that it is not left blank by the user.



5. Add a set of radio buttons to the form to accept a level of entry such as GCSE, AS or A2. Write a function that displays the level of entry to the user in an alert box so that the level can be confirmed or rejected.

Plan for Code

- Heading into the task, I decided to do research on radio buttons using web sites such as 'w3schools.com' and 'tutorialspoint.com'. These helped me to understand the code that was associated with radio buttons, in both HTML and JavaScript.

My plan is:

1. Create five radio buttons: 'GCSE', 'BTEC', 'AS', 'A2' and 'PHD'.
2. Make sure the user checks a radio button.
3. Identify which radio button is checked.
4. Display the level of entry selected in an alert box so that it can be confirmed or rejected by the user.

1- The radio buttons will be all on one line so that the layout looks clear and neat to the user.

2- To make sure a radio button is checked, at least two things can be done:

1) An IF statement can detect if each radio button has been checked or not.

2) One of the radio buttons are created to be checked from the start of when the form is run. Then, the user can change which radio button is checked, or keep the chosen entry level checked if it is correct.

- The second option is much easier and more efficient, so I will choose this.

3- To identify which radio button has been checked and thus which entry level has been selected, I must use IF statements.

- To make IF statements for all five radio buttons it would take much time and the code would be long and inefficient. But I could use a FOR LOOP to make one IF statement that would be repeated for each of the radio buttons. This would same time and make my code more efficient.

4- To display the entry level that the user has checked in an alert box:

- will display the selected entry level, depending on which radio button is checked, every time the user clicks the 'submit' button, so that they can confirm or reject it.
- The confirm method will contain the value that is returned from the new function that I will create, after the function detects which radio button the user has checked.

Written Code

```
27  var conf = confirm(function2());
28  if (conf == false)
29  {
30      result = false;
31  }
```

- In **Line 28** the condition of the IF statement is if 'conf is false'.

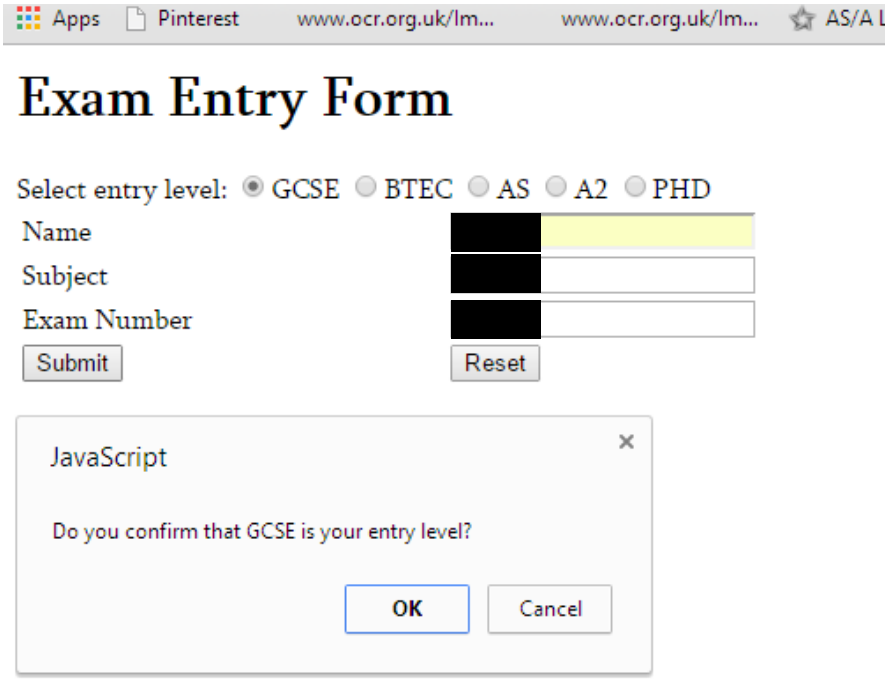
This means that if the user rejects whatever is in the alert box to confirm, then 'result' is false, and thus the form is invalid, so the success page is not loaded and instead the user can change their selected entry level.

- 'whichButt' is basically an array of the values of each individual radio button in the group. Because all of the five radio buttons have the same name ('level'), they are part of a group and with the method: 'document. getElementsByName()', these values are stored in 'whichButt'.

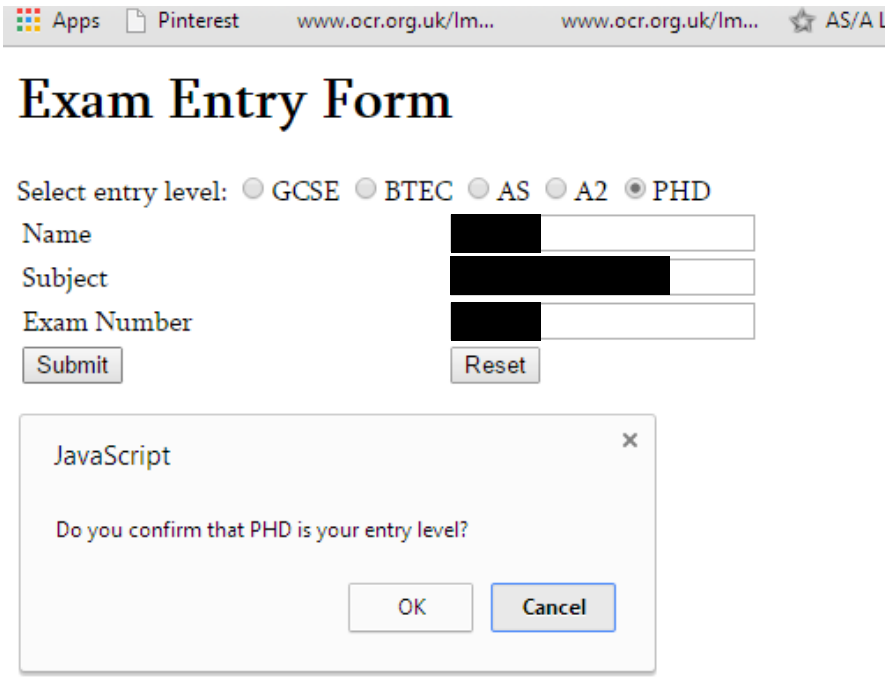
```
40  function function2()
41  {
42      var whichButt = document.getElementsByName("level");
43      var leng = whichButt.length
44
45      for (i = 0; i < leng; i++)
46      {
47          if (whichButt[i].checked)
48          {
49              return "Do you confirm that " + whichButt[i].value + " is your entry level?";
50          }
51      }
52  }
```

- In **line 42** the variable 'whichButt' is declared and initialised. The variable: 'leng' is equal to the length of this group of radio buttons so it is equal to how many radio buttons share the same name and in this case it is 5, so 'leng' is 5.
- In **line 45** the FOR LOOP iterates once for each value of 'i'. The value of 'i' is 0 to begin with then increases by 1 after every iteration. The FOR LOOP stops iterating when the value of 'i' reaches one less than the value of leng. Since leng is 5, so it iterates, or cycles, 5 times.
- On **line 47** the IF statement checks which radio button has been checked. As the FOR LOOP iterates, the IF statement checks if each radio button in the position of i's current value in the array: 'whichButt', is checked. When i reaches a value equal to the position of the checked radio button within the array, then the confirmations statement containing the chosen entry level is returned to the variable 'conf' so that the alert can be displayed to the user.
- **Line 49** contains the message that is displayed in the alert to the user so that they may confirm or reject their chosen entry level. 'whichButt[i].value' is the actual value of the radio button that has been checked, and this is displayed here as is required by the task.

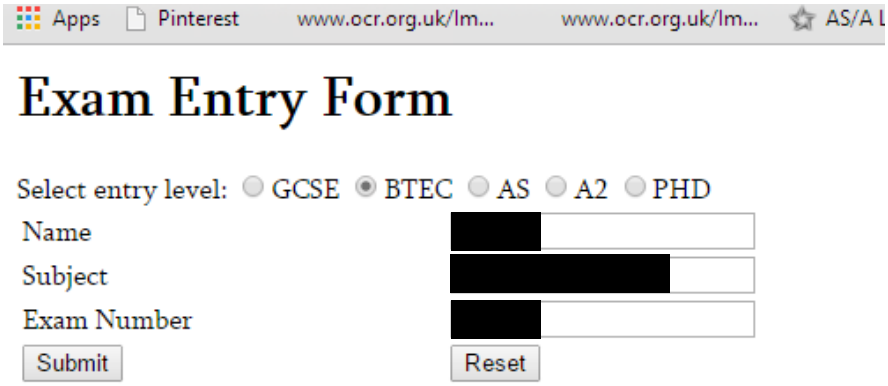
TESTING SOLUTIONS



This screenshot is of all the fields being filled correctly and the user has pressed 'submit'. Before the 'success' page is loaded, this JavaScript alert is displayed, with the message: "Do you confirm that GCSE is your entry level?" Then the user can press OK to confirm this or 'cancel' to reject this.

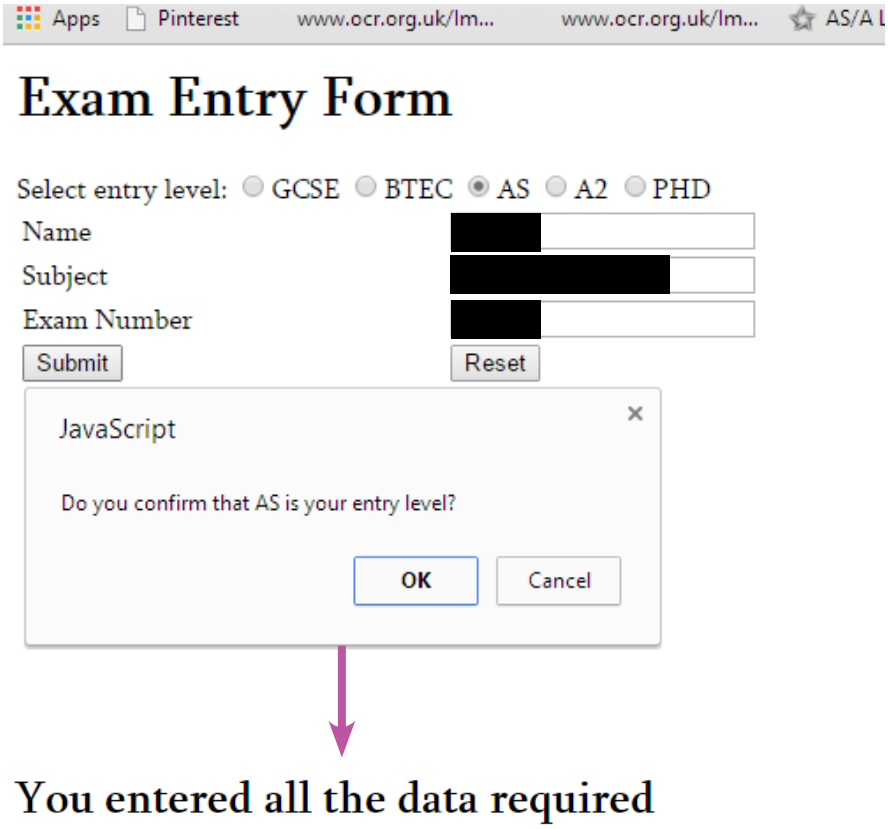


The user then incorrectly selected 'PHD' as their entry level, and this screenshot shows that the prompt is displayed to confirm if the correct entry level has been selected. The user is given a chance to change it to the correct one. Here, he is shown to click 'Cancel' to go back to the form and check the correct radio button.



The page returns to the form so that the user can now change the selected entry level radio button:

Now AS, the correct entry level of the user is selected after the user has been given the chance to check the correct radio button.



This screenshot shows that when the radio buttons are correctly selected and confirmed the success page is loaded.

6 – Produce an evaluation of your solutions

Solution 1 – 'Add another text field to the form to take the user's examination number'.
Finding the solution to this was relatively straight forward as the code for taking the name and subject of the user was already provided. The same code for the text field is used: **'<input type="text"…** To display the text field's name, a new cell in the table is created with the code: **<td id="exam">Exam Number</td>**, which follows as similar structure of the names of the other two text fields that are displayed.

So it is a combination of observing the code that was provided for the HTML file as well as the knowledge of HTML that I had obtained at school.

Solution 2 – 'Extend the JavaScript code to validate this field to make sure that it is not left blank.
As with the solution to the previous task, I observed the code provided to create a solution to this task. Therefore, it was not a particularly challenging one. Ensuring that the text fields are not left blank is essential to any form, and this can simply be validated if the value that the user inputs into the text field is equal to "" a blank space.

Solution 3 – 'Extend the JavaScript code to make sure that the user's examination number is exactly 4 digits.
So the requirement in this task also means that the input of the user in the 'exam number' text field is a number, which is then also 4 digits long. JavaScript contains a useful method which is **isNaN().** This checks if a variable is a number. The . length method is also very useful to check if the value that the user enters is 4 characters long. So these two methods used together in an IF statement ensure that it the text field is validated so that the exam number is a 4 digit number.

Solution 4 – Add a set of radio buttons to the form to accept a level of entry such as GCSE, AS or A2. Write a function that displays the level of entry to the user in an alert box so that the level can be confirmed or rejected.
This task proved slightly more time-consuming as it required a completely new element to the web page. I knew little about radio buttons and so I researched them on the internet, using websites such as **w3schools.com** and **tutorialspoint.com** to get to know about them. . The way that JavaScript works with functions was also new to me and so I did some research on the internet about functions, and I learned how to call functions and return values so that I could create a solution to the task. I learned how radio buttons can be effectively grouped together if they are all given the same 'name' value, and that the number of radio buttons in this 'group' could be found using the **.length** method. I then discovered that each radio button in the 'group' effectively had an ordinal position within the 'group' and these could be used so that each radio button can be referred to be checked if it has been checked by the user. This is done efficiently within a 'FOR loop' and for a larger set of radio buttons, using a 'FOR loop' is definitely the most efficient method, as code does not have to be repeated. However, the task only requires 3 radio buttons as a minimum, and 'IF statements' can be used for each of these to check if they have been checked by the user as 3 is a small number of radio buttons. When the code is run and the web page is loaded, I ensured one of the radio buttons is checked, meaning that I do not have to write code to deal with the situation where the user does not check any radio button. This could, of course, be validated, perhaps using a variable to store the number of checked radio buttons as the 'FOR loop' iterates and if the value if equal to 0 (meaning that 0 radio buttons have been checked) then using an 'IF statement' to detect this, an appropriate message within an 'alert', perhaps, can be displayed. However, within the 'Google Chrome' web browser, there was an issue with this as the browser would not load the radio buttons with one checked. But 'Internet Explorer' would, and this was due to differences within the browsers themselves.

7. Write a conclusion about the effectiveness of JavaScript validation routines to reduce the number of errors that are made in data input.

JavaScript validation routines have some benefits and drawbacks at the same time, regarding the effectiveness of reducing errors. Using JavaScript is an example of a client-side form validation, where data inputted by the user is not sent to a server during the validation. One of the main advantages is that forms that must be validated run code on the user's computer without having to send the data inputted by the user across networks and then produce a response back to the user. That this back and forth sending of the data inputs does not take place with JavaScript validation routines, time and bandwidth is saved and for larger forms, it is likely that the process will be faster as well. But one disadvantage of using this 'client-side' validation is that security is decreased. Users can enter unwanted data to the server or dangerous data. They can even use programs that can mimic the response and submission of the form. But for forms where a server-side validation is needed, perhaps when someone wants to know about the data that a user is entering, then JavaScript validation routines might not be as useful. During a JavaScript validation routine you will not be able to see the data that the user entered, so what errors they made or how often an error is made and this means that JavaScript validation routines can sometimes not be the most useful way of trying to analyse the data that is inputted by users and their errors so it will be difficult to see how and where errors are made to reduce the number of errors made in data input.

The small print

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

OCR Resources: *the small print*

OCR's resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

OCR acknowledges the use of the following content:

Square down and Square up: alexwhite/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: resources.feedback@ocr.org.uk

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

ocr.org.uk/alevelreform
OCR customer contact centre

General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk