Chapter # 19 – Programming for the Web

Specimen 2017

5 In computer programming, explain what is meant by the following terms.

(a) variable

....

....

....

.... [2]

Answer:

5(a)	Award 1 mark for each correct answer up to a maximum of 2. **Two** from: A storage location A symbolic identifier Holds a quantity/value/information The variable name is used to refer to the stored value	**2**

(b) iteration

....

....

....

.... [2]

Answer:

5(b)	Award 1 mark for each correct answer up to a maximum of 2. **Two** from: Repetition of a process/block of code/statements In order to meet a wanted result Results of one iteration used as starting point for the next	**2**

Question 5, Part (c)

(c) loop

....

....

....

.... [2]

Answer:

5(c)	Award 1 mark for each correct answer up to a maximum of 2. **Two** from: Repeating a set of instructions/statements Until a pre-defined condition is met/pre-set number of loops carried out, e.G. If ... then ... else	2

June 2017 – P31 & P33

3 Frank is a web designer who writes his own code and uses JavaScript on his webpages. He wants to print his name on every page. He creates two versions of the code to run his JavaScript on the webpage. These versions are represented here:

Version 1:

```
<script language="JavaScript">
document.write ("<p>My name is <b>Frank </b> </p>");
</script>
```

Version 2:

```
<script language="JavaScript" src="FranksJavaScriptcode.js">
</script>
```

where the instructions for writing Frank's name are in the file called "FranksJavaScriptcode.js"

Explain why Frank prefers to use Version 2 of the code.

....

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

Question 3

Answer:

| 3 | **Six** *from:*

Can call the code several times/from different pages/re-use the code
No need of re-writing/having several copies/copies on each webpage
Code only has to be tested once/checked for errors once
File/JavaScript is cached by web browser
...reduces network access time/reduces cost of fetching data
JavaScript code embedded in webpages can slow loading times/reduce browser performance
...webpage can slow/stop while browser executes code
Can separate code into different conceptual/functional areas
...provides modularity to code
...separate html and JavaScript code so easier to read/maintain. | 6 |

June 2017 – P32

9 JavaScript is a programming language used in webpages.

Fig. 2 shows a table created with JavaScript in a webpage.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Fig. 2

The code that created the table is shown below:

```
<html>
   <body>
      <script language="javascript">
         var tableout;
         tableout = "<table border='1' width='300' cellspacing='0'
         cellpadding='3'>"
         for (b = 1; b <= 10; b++) {
            tableout = tableout + "<tr>";
            for (g = 1; g <= 10; g++) {
               tableout = tableout + "<td>" + b * g + "</td>";
            }
            tableout = tableout + "</tr>";
         }
         tableout = tableout + "</table>";
         document.write (tableout);
      </script>
   </body>
</html>
```

Question 9 (Continued)

Answer:

Explain how the loops in the code create the table.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[8]

Question 9

Answer:

| 9 | **Six from:**

Code is embedded within the body HTML code
(global) variable (tableout) is declared/created/initialised
The HTML table values are placed within the variable
First/outer loop (on 4th line of JS code) executes 10 times…
…to create 10 rows using global variable and HTML <tr> code/to create each row
Second/inner loop is executed each time outer loop executes
…to create 10 columns/cells
First time inner loop executes, the cell contains 1*1=1...
…second time inner loop executes, the cell contains 1*2=2
…third time inner loop executes, the cell contains 1*3=3
…up to cell that contains 1*10=10
When inner loop reaches 10, first row of cells is complete…
…next row is started with 2*1=2, 2*4 etc.
…up to 2*10=20
The process continues until outer loop reaches 10 and all 10 rows have been created and filled. | 8 |

November 2017 – P32

11 A shop is open between 12 noon and 10 pm.

While the shop is closed in the morning, a message saying "Sorry, we are closed" is displayed. After the shop has closed in the evening a message saying "Please try again tomorrow" is displayed. At all other times, a message saying, "Hello, we are open now" is displayed.

JavaScript code can be used to alter web pages in real time.

Complete the code in the function below that will allow a user to find out if the shop is open or closed.

```
function OpenTimesFunction() {
var welcome;
var timenow = new Date().getHours();
```

....

....

....

....

....

....

....

....

```
document.getElementById("open").innerHTML = welcome;
//This line displays the result of the code
}
```

[6]

Question 11

Answer:

11(b)	**Six** *from:*	6
	An example code is:	
| | ```
if (timenow <12) {
   welcome = 'Sorry, we are closed';
} else if (timenow < 22) {
   welcome = 'Hello, we are open now';
} else {
   welcome = 'Please try again tomorrow';
}
``` | |
| | *1 mark per correct line.* | |

March 2017 – P32

9   JavaScript can be embedded in the code of web pages to add interactivity to a page.

Jonas uses JavaScript code to store, list and display the names of cities on a web page.

(a)   Two methods of storing the names are shown below.

Method 1:

```
var city1 = "London";
var city2 = "Cambridge";
var city3 = "Oxford";
var city4 = "Manchester";
```

Method 2:

```
var city = ["London", "Cambridge", "Oxford", "Manchester"];
```

Explain why storing the names using Method 2 is more suitable for storing large numbers of cities.

....

....

....

....

....

....

....

....

....

....[4]

Question 9, Part (a)

Answer:

9(a)	**Four** from: Method 2 uses an array... ....which stores multiple values in a single variable More suitable for storing large numbers of items/data items as it reduces the complexity of the code Increases the code easier to understand Increases the execution speed of the code Method 2 can be looped through using an iterative function ...to find a specific data item	4

Question 9, Part (b)

(b) Jonas has added some lines to the JavaScript code:

```
/*
The code below stores the list of cities.
*/
var city = ["London", "Cambridge", "Oxford", "Manchester"];
```

Explain why Jonas would have added the new code to the script.

....

....

....

....

....[2]

Answer:

9(b)	**Two** from: Jonas wanted to explain/add comments to the code/what the line of code means/is for To make it clear that the code referred to a list of the cities Ensures that the explanation/comment was ignored by the web browser To make the code more readable/understandable	2

Question 9, Part (c)

(c) Jonas wants to extract 'Oxford' from the list in the code below to display it on the web page.

```
var city = ["London", "Cambridge", "Oxford", "Manchester"];
```

Write a line of JavaScript code to access the list and store the city name.

....

....

....

....

....

....[3]

Answer:

9(c)	A suitable line of code would be: var place = city(2)	3
	Three *from:*	
	var =1 mark plus suitable variable name to store city e.g. place =1 mark = city(2) =1 mark	

Question 9, Part (d)

(d) Write a loop in JavaScript code to extract and display the first three city names.

....

....

....

....

....

....

....

....[4]

Answer:

| 9(d) | Suitable code would be:

for (b = 0; b <= 3; b++) {
       document.write (city (b));
    }

Marks, **four** *from:*

for ()   1 mark
suitable var names   1 mark
count from to 0 to 2 (b from 0 to <=3)   1 mark
adding 1 inside loop (b++)   1 mark
displaying the result of loop   1 mark | 4 |

June 2018 – P31

2 JavaScript defines a number of primitive data types.

(a) Explain the term 'primitive' when used in this context.

....

....

....

....

....[2]

Answer:

2(a)	**Two** *from:*	2
	(The data type is) hard-coded/built-in Cannot be altered/is fixed Have no additional properties.	

Question 2, Part (b)

(b) Describe **three** primitive data types used in JavaScript.

1 ....

....

....

2 ....

....

....

3 ....

....

....[3]

Answer:

2(b)	**Three** *from:*	**3**
	String is a series of characters Number is any number, with or without decimal places Boolean has only two possible values, true or false Undefined is a variable without a value Null is 'nothing' but it is still an object in JavaScript, it is usually empty Symbol has a unique identifier, is static.	

3   Variables are used in JavaScript to hold values.

Explain how a variable is created in JavaScript code.

....

....

....

....

....

....

....

....

....[4]

Answer:

3	**Four** *from:* Use a (suitable) name …. that is not a reserved word Declare the variable with 'var' (command word) Declared once only in the script/code Use as global or local variable but not both Initialise the variable with a value Do not use quotes around the variable name.	4

June 2018 – P32

4   'Loops' are used in JavaScript to execute a block of code several times.

The following code shows a 'for' loop and the code for displaying the result.

```
<p id="Number"></p>

    <script>
 //JavaScript code follows
        var displayresult = "";
        var X;

    for (X = 1; X < 10; X += 3) {
    displayresult += X + "<br>";
    }
//the following line displays the contents of 'displayresult' on the
webpage
    document.getElementById("Number").innerHTML = displayresult;
</script>
```

(a) Describe what each statement in the 'for' loop does and the results that are output when the code is executed.

....

....

....

....

....

....

....

....

....

....

....

....[5]

Question 4, Part (a)

Answer:

4(a)	**Five** *from:* Variables X and 'displayresult' are declared ...and cleared before use by loop Loop starts with X at 1 X is incremented by 3 each time it loops Continues until X reaches 10/while X is less than 10 Displays result as 1, 4, 7 With carriage return between each/on separate lines/underneath each other.	5

Question 4, Part (b)

(b) Rewrite the 'loop' code as a DO WHILE loop to display the same results.

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

Question 4, Part (b)

Answer:

4(b)	*Suitable code could be:*		6
	`<p id="Number"></p>` `/JavaScript code follows` `<script>` `var X = 1;` `do {` `    document.getElementById("Number").innerHTML += X + " ";` `    X=X+3` `}` `while (X < 10)` `</script>` var X =1; 1 mark Do {...} 1 mark +=X 1 mark + " " 1 mark X=X+3 1 mark while (X < 10) 1 mark		

Question 4, Part (c)

(c) The JavaScript code is deemed to be 'client-side' code.

Explain why the use of 'client-side' code can cause problems.

....

....

....

....

....

....

....

....

....[4]

Answer:

4(c)	**Four** *from:* The code is executed by the web browser Not on the web server Web browser may not support the code language So the code may not execute properly/at all/produce errors Different browsers run code in different ways Developers must test all code with all browsers Same browsers on different operating systems behave differently Code may produce different results Code requires high processing power So webpages may display slowly/not at all Non-functioning code may deter viewers leading to loss of audience/sales via the website.	4

June 2018 – P33

2   The use of JavaScript within the HTML code of a web page allows the page to react to user intervention.

The code below contains a function named **_checkreadpagefunction_** that will ask the user to confirm that the page has been read.

Complete the JavaScript code by writing extra code that will capture the click event and execute the function.

```
<html>
<body>
<p>Click on the button to confirm that you have read this page.</p>
<button id="button1">Click here</button>
<script>
```

....

....

....

....

....

....[6]

```
function checkreadpagefunction() {
    alert ("I have read this page.");
}
</script>
</body>
</html>
```

You may use the space below for any rough work.

Question 2

Answer:

| 2 | *A suitable line of code would be:*

document.getElementById("button1").addEventListener("click", checkreadpagefunction);

Six *from:*
1 mark for each of:

Capturing the element:
    document.getElementById
Identifying the button by name:
    "button1"
Checking for the event:
    addEventListener
Correct reference to click by mouse:
    "click",
Calling the correct function by name:
    checkreadpagefunction);
All correct delimiters and all correct brackets:
    . between key words, , after click AND ; after function
    () around button1
    () around parameter ("click", checkreadpagefunction); | 6 |

November 2018 – P32

5 Quintin is a programmer who writes code in HTML and JavaScript for use in online forms. The code is developed and 'white-box' tested before being used.

Quintin is developing the code below. He has added comments to the code. The code asks a user to input two numbers and adds the numbers together. It also displays the total and whether or not it is greater than 10.

```html
<html>
<body>

<!-- the next six lines collect the two numbers to be added-->

<br/>Enter first number:
<input type="number" id="nm1" name="num1">
<br/><br/>Enter second number:
<input type="number" id="nm2" name="num2">
<p id="add"></p>

<script>
  function myaddfunction() {
//the next line assigns the first number input to the variable y
            var y = document.getElementById("nm1").value;
//the next line assigns the second number input to the variable z
            var z = document.getElementById("nm2").value;
//the next line adds the two numbers and assigns the result to the variable x
            var x = +y + +z;
//the next line checks if x is greater or not greater than 10 and reports
accordingly
            var A = (x >10) ? x+ " is greater than 10":x+ " is not greater
than 10";
//the next line prints the results onto the page
document.getElementById("add").innerHTML = A ;
    }
</script>
<!-- the next two lines asks the user to click the button and then the script
is executed
<p>Click the button to calculate the total.</p>
<button onclick="myaddfunction()">Add the numbers</button>
<br/>
  </body>
</html>
```

Question 5, Part (a)

(a) Explain how Quintin could use 'white box' testing to ensure that the JavaScript code produces the correct result every time.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[8]

Question 5, Part (a)

Answer:

| 5(a) | **Eight** *from:*

Checking each line of code/statement
Ensures that each line of code is executed at least once
Ensures that var y and z assign the collected numbers as required
Checks that the additon of y and z is correct
Ensures that the correct message is displayed when result assigned to A
Checking each branch/decision in the code
Checks that decisions are carried out correctly
So that values put in A can be compared
Ensures that result is checked against > 10
Ensures that the correct result is put in var × as required
Checks every possible pathway through the code
So that test values in var y and z cause each subsequent path to be followed
So that test values in x are assigned to A to produce both the messages "is greater than 10" and "is not greater than 10" depending on value in A. | 8 |

Question 5, Part (b)

(b) Explain why it is good practice that Quintin places his JavaScript code in external files rather than embeds the code in the actual page code.

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

Answer:

5(b)	**Six** *from:*	6
	Can separate code into different conceptual/functional areas for ease of development/testing/understanding Separating HTML and JavaScript code provides modularity to code Which is easier to read/maintain/update by Quintin/different coders as required Can call the code several times/from different pages/re-use the code No need to rewrite/have several copies/copies on each web page Code only has to be tested once/checked for errors once File/JavaScript is cached by web browser No need to reload it/fetch code repeatedly if need on other pages Reduces network access/reduces cost of fetching data JavaScript code embedded in web pages can slow loading times/reduce browser performance Web page can slow/stop while browser executes code.	

November 2018 – P33

5   JavaScript can be embedded in the code of web pages to add interactivity to a page.

Explain what is meant by the following terms when they are used in JavaScript:

(a) an array.

....

....

....[1]

Answer:

5(a)	Stores multiple values in a single variable.	1

(b) a variable.

....

....

....[1]

Answer:

5(b)	Containers for storing data values.	1

(c) a function.

....

....

....[1]

Answer:

5(c)	**One** *from:* A block of code designed to perform a particular task Code executed when it is invoked (called).	1

Question 5, Part (d) and (e)

(d) a comment.

....

....

....[1]

Answer:

5(d)	**One** *from:*	1
	Text preceded by // is not executed/ignored by JavaScript Used to explain the code Used to halt execution of the code Text that is not executed before a line of code Text that is not executed at end of line of code.	

(e) an object.

....

....

....[1]

Answer:

5(e)	**One** *from:*	1
	A collection of variables and functions Representing the attributes and behaviour of an 'item'/'thing' being modelled in a program.	

Question 5, Part (f)

(f) an expression.

....

....

....[1]

Answer:

5(f)	**One** *from:*	1
	Any (valid) unit of code that resolves to a value Two types of expression exist: ....can have a value ....can assign a value to a variable.	

March 2018 – P32

6   Drivers who are 16 years of age or older can apply for a driving licence online.

The government web page code requires applicants to enter their age and then checks if they are old enough to apply.

The following code captures the applicant's age. When the applicant clicks the 'Check now' button the age is first checked to ensure that it is numeric. The age is then checked to see if the applicant is old enough. Suitable messages are displayed as a result of the checks.

Complete the code for the function **CheckAgeFunction** that checks the age entered by the applicant and displays an appropriate response.

[Question 6 Continues On The Next Page]

```
<html>
<body>
<p>You can apply for a licence to drive when you are 16 years old.</p>
<p>To check if you are old enough to drive, input your age and click
the button:</p>
<input id="AgeNow" value="16" />
<button onclick="CheckAgeFunction()">Check now</button>
<p id="AgeCheck"></p>
<script>
        function CheckAgeFunction() {
```

....

....

....

....

....

....

....

....

....

....

....

....[8]

```
    }
    document.getElementById("AgeCheck").innerHTML = CanApply;
}
</script>
</body></html>
```

Question 6

Answer:

| 6 | *Eight* from:

Suitable code would be:

var AgeNow, CanApply;
    AgeNow = Number(document.getElementById("AgeNow").value);
    if (isNaN(AgeNow)) {
        CanApply = "Please enter your age in numbers.";
    } else {
        CanApply = (AgeNow < 16)? "You are too young to apply for a licence.": "You are old enough to apply for a licence.";

Mark points: 8 from:

Declare the variables, must be exact variable names as in Question:
           var AgeNow, CanApply;

Capture the input of the age:
           AgeNow=Number()

Use of correct capture code:
    document.getElementById("AgeNow").value;

Use of "isNaN" to check that the input is a number:
                if (isNaN(AgeNow))

Display error message if not a number:
    CanApply = "Please enter your age in numbers.";

Use of "if...else" to make decision:
                } else {

Use of comparison check: CanApply = (AgeNow < 16)?

Appropriate display messages: e.g. "You are too young to apply for a licence.": "You are old enough to apply for a licence.";

Messages match comparison: i.e.:
                < 16... too young; old enough
                >16.... old enough; too young | 8 |

June 2019 – P31 & P33

2 In JavaScript code, the sort() function is used to sort lists into ascending order.

(a) Explain, in detail, why, when using the sort() function, the list in Fig. 2.1 is sorted correctly, but the list in Fig. 2.2 is not.

Before sorting	After sorting	Before sorting	After sorting
fly	ant	1345	1111
cockroach	beetle	3666	12
ant	butterfly	1111	1345
butterfly	cockroach	23	23
moth	fly	37	3666
beetle	moth	12	37

| Fig. 2.1 | Fig. 2.2 |

....

....

....

....

....

....

....

....

....

.... [4]

Question 2, Part (a)

Answer:

2(a)	**Four** *from:*	4
	Function sort () treats values as strings not numbers Strings are sorted alphabetically Strings are not sorted numerically a is before / 'lower' than b so list 1 is sorted alphabetically by the first letter and then by the second etc. in list 2, the list is also sorted alphabetically so e.g. 1111 is before 12 because 2 is 'bigger' than 1 *Max. 1 for additional examples e.g.:* The third character in 1111 has no match so is 'bigger' than 12 3666 is before 37 because while the 3s match, and 6 is before 7, there is no match for the second 6 so it is 'bigger' than no number.	

Question 2, Part (b)

(b) Write a line of JavaScript code that could be used to sort the list of insects in Fig. 2.1 into descending order.

....

....

.... [2]

Answer:

2(b)	A suitable line with the function is: insects.reverse(); The variable name can be anything suitable, reverse() is the function. *Mark allocation:* Use of suitable variable name e.g. insects 1 mark All correct function and syntax .reverse(); 1 mark	2

3 An online retailer uses a simple form on its website to enable customers to contact its After Sales department by email. The form looks like this:

Send an email to aftersales@mycompany.com:

Your Name:

Your email address:

Comment:

Send   Reset

Some of the code that created the form is shown.
Note that the lines have been numbered only for your convenience and reference.

```
1   <html>
2   <body>
3
4   <h2>Send an email to aftersales@mycompany.com:</h2>
5
6   <form action="mailto: aftersales@mycompany.com" method="post"
    enctype="text/plain">
7
8   Your Name:<br>
9
10  Your email address:<br>
11
12  Comment:<br>
13
14
15
16  </form>
17  </body>
18  </html>
19
20
```

Question 3, Part (a)

(a) Explain what the different parts of the code in line 6 do.

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

Question 3, Part (a)

Answer:

| 3(a) | **Six** *from:*

<form action="mailto: aftersales@mycompany.com"
Tells the page that this is a form to be actioned by sending / submitting the form to a specified URL / default URL is this page
To send an email via mailto to the specified address

method="post"
Specifies the HTTP method to be used when submitting the form
In this case post means not to display the submitted data / used for sensitive or private / personal data / make the submitted data invisible in the field / will not allow bookmarking / is not saved in browser history
Post can send unlimited amounts of data so no need to specify the size

enctype="text/plain">
Specifies the encoding of the data
As plain text. | 6 |

Question 3, Part (b)

(b) Create some additional lines of code that could be inserted into the script at appropriate locations to collect the name and email address of the customer. Indicate, with reference to the line numbers, where your additional code should be inserted.

Line number	Code

[4]

Answer:

3(b)	**Four** *from:*			4
		Line number	**Code**	
		9	`<input type="text"` 1 mark `name="name">` 1 mark ` ` 1 mark	
		11	`<input type="text name"="email">  ` 1 mark	

Question 3, Part (c)

(c) Write down the code that would allow:

- the comment to be entered
- the form to be submitted
- the form to be reset.

Indicate which line numbers the codes would appear on.

Line number	Code

[6]

Question 3, Part (c)

Answer:

3(c)	Line number	Code	6
	13	`<input type="text" name="comment" size="100">`	
	14	`<input type="submit" value="Submit your details">`	
	15	`<input type="reset" value="Reset the form">`	

Six *from:*

Code on correct lines
Correct syntax
Size="100" or suitable value
value="Submit your details" following submit
value="Reset the form" following reset
Correct input types   1 mark each

June 2019 – P32

3   An author has written the source code of a web page that will be used when a person applies for a driving licence. The code, shown in Fig. 3.1, is intended to check that a person is at least 16 years of age.

The lines of the code are numbered only for your convenience when referring to the code.
The JavaScript code is in lines 8 to 19.

```
1    <html>
2    <body>
3    <p>You can apply for a licence to drive when you are 16 years
     old.</p>
4    <p>To check if you are old enough to drive, input your age and
     click the button:</p>
5    <input id="AgeNow" value="16" />
6    <button onclick="CheckAgFunction()">Check now</button>
7    <p id="AgeCheck"></p>
8
9          function CheckAgeFunction()
10   var AgeNow;
11       AgeNow = Number(document.getElementById("AgeNow").value);
12       if (isNaN(AgeNow)) {
13           CanApply = "Please enter your age in numbers.";
14       } else {
15           CanApply = (AgeNow >15)? "You are too young to apply
     for a licence.": "You are old enough to apply for a licence.";
16       }
17       document.getElementById("AgeCheck).innerHTML = CanApply;
18   }
19
20   </body>
21   </html>
```

Fig. 3.1

Testing has shown that the code contains a number of errors of different types which must be corrected before the code will perform as expected.

Question 3 (Continued)

Identify the line numbers containing the errors. Explain why each of the errors prevents the code from running correctly and how each should be corrected. Use the table for your response.

Line number of error and explanation of error.	Explanation of suggested correction.

[8]

Question 3

Answer:

3			8
	Line number of error and explanation of error	**Explanation of suggested correction**	
	Line 6/9 and the function is spelt incorrectly / differently, so will not run on button click	Should be same as function/ CheckAgeFunction/ CheckAgFunction	
	Line 8 script is not opened so web browser cannot interpret it	Add <script> to open the script	
	Line 9 missing {/open curly bracket so line is not terminated correctly	Add {/open curly bracket	
	Line 10 the variable CanApply is not declared so cannot be used in function	Add , to separate variables / add new line with var / add CanApply to declare the variable	
	Line 15 age is wrongly shown as 15 so age check is incorrectly compared	Should be 16 as per intended age check/stem/line 3	
	Line 15 incorrect logic comparison so age messages are reversed when displayed	Change to < for correct comparison/reverse the messages to match comparison	
	Line 17 " is missing from ("AgeCheck " so AgeCheck is not interpreted as p id so its value is not returned	Add "	
	Line 19 script is not closed so web browser cannot interpret it	Add </script> to close the script	
	1 mark for error and 1 mark for <u>matching</u> correction.		

Question 3

Answer (Continued):

3 | The errors are shown here:

```
1    <html>
2    <body>
3    <p>You can apply for a licence to drive when you are 16 years
     old.</p>
4    <p>To check if you are old enough to drive, input your age and
     click the button:</p>
5    <input id="AgeNow" value="16" />
6    <button onclick="CheckAgeFunction()">Check now</button>
7    <p id="AgeCheck"></p>
8
9         function CheckAgeFunction()
10   var AgeNow;
11        AgeNow = Number(document.getElementById("AgeNow").value);
12        if (isNaN(AgeNow)) {
13            CanApply = "Please enter your age in numbers.";
14        } else {
15            CanApply = (AgeNow < 16)? "You are too young to apply
     for a licence.": "You are old enough to apply for a licence.";
16        }
17        document.getElementById("AgeCheck").innerHTML = CanApply;
18   }
19
20   </body></html>
21
```

November 2019 – P31 & P33

11 The area of a rectangle can be calculated using the JavaScript embedded in a web page as shown below.

```
1  <html>
2  <body>
3  <script>
4  var length = parseFloat(prompt("Enter length of the rectangle : "));
5  var width = parseFloat(prompt("Enter width of the rectangle : "));
6
7  var calc_area = (length * width);
8
9      document.write("<br>");
10     document.write("<h3> Area of a rectangle</h3>");
11     document.write("<font face='arial' size='3'>")
12     document.write(" The sides of the rectangle are " + length + " by " +
       width + ".</font><br>");
13     document.write("<font face='arial' size='3'>")
14     document.write(" The area of the rectangle  is " + calc_area +
       ".</font><br>");
15
16 </script>
17 </body>
18 </html>
19
```

Question 11 (Continued)

Explain, with reference to the code shown, how the JavaScript collects the dimensions of the rectangle, calculates the area and displays the area on-screen.

The lines of the code are numbered only for your convenience when referring to the code.

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

Question 11

Answer:

11	*Six from:* Line 4 declares the variable/var length to hold one side of rectangle Line 5 declares the variable/var width to hold other side of rectangle parseFloat (prompt("")); used to display message parseFloat (prompt("")); used to collect values for both sides of rectangle parseFloat (prompt("")); used to create a (text) box for the user to enter the values Variable/var calc_area is declared to calculate the area Holds result of calculation document.write() is used to display the messages on screen about the values/area of the rectangle Displays the results of the calculation/contents of var calc_area.	6

November 2019 – P32

3 JavaScript uses strings to store data.

This script is intended to display the contents of three variables on a web page.

Line numbers are provided only for your convenience when referring to the code.

```
1   <html>
2   <body>
3
4   <p id="names"> </p>
5   <script>
6   var statementtxt1 = "It's only me, 'Hardeep'";
7   var statementtxt2 = "His name is "Peter"";
8   var statementtxt3 = "We call her Jasmine";
9   document.getElementById( 'names' ).innerHTML = statementtxt1 + "<br>"
    + statementtxt2 + "<br>" +  statementtxt3;
10  </script>
11  </body>
12  </html>
```

Explain, giving reasons, why no output is produced when this JavaScript is executed by a web browser as part of a web page.

....

....

....

....

....

....

....

.... [3]

Question 3

Answer:

3	**Three** *from:*	3
	Line 7 contains a syntax error "Peter" is enclosed in quotes that are the same as the quotes for the string (Strings in JavaScript can contain quotes but) the quotes in a string must not be the same as the enclosing quotes Any syntax error causes the script to fail/not run No error message is produced.	

4 A web page uses JavaScript code to display a list of food crops.

Line numbers are provided only for your convenience when referring to the code.

```
1   <html>
2   <body>
3   <p id="foodcrops"></p>
4   <script>
5   var crops = ["corn", "rice", "maize", "sugarcane"];
6   var i = 0;
7   var show = "";
8   while (crops[i]) {
9       show += crops[i] + "<br>";
10      i++;
11  }
12  document.getElementById("foodcrops").innerHTML = show;
13  </script>
14  </body>
15  </html>
```

(a) Explain, in detail, how the JavaScript code produces this list from an array.

corn
rice
maize
sugarcane

....

....

....

....

....

....

....

Question 4, Part (a) (Continued)

....

....

....

....

....

....

....

....

....

....

....

.... [8]

Answer:

4(a)	**Eight** *from:* Code is embedded in HTML code of the website by <script> and </script> delimiters/markers The browser executes the JavaScript code within the delimiters Variable crops is set to contain the list of crops Variables i and show are initialised While loop will continue looping as long as crops[i] contains data/string (var) i is incremented by 1 each time loop is executed (var) show is set to the current value of show concatenated with next crop value Loop terminates when array has no more items/final value in array is reached Final values of (var) show are displayed on screen/web page Values shown in vertical list as code forces carriage return/line feed.	8

Question 4, Part (b)

(b) Explain how you would amend the existing JavaScript code to add 'beans' to the array so that it would produce this list.

beans
corn
rice
maize
sugarcane

....

....

....

....

....

.... [2]

Answer:

| 4(b) | **Two** *from:*

Amend the array `var crops = []` **1 mark**
to `["beans", "corn", "rice", "maize", "sugarcane"];` **1 mark**

Add the line
crops.unshift("beans"); … **1 mark**
… any line between var crops and while() **1 mark**

Amend the var show line to … **1 mark**
… `"beans <br>";` **1 mark** | **2** |

March 2019 – P32

1   Errors in JavaScript code can cause the code to fail to execute when run by a web browser. The code shown is intended to display a table on a web page when run in a web browser.
The line numbers are shown only for your convenience when referencing the code in your answers.

```
1   <html>
2   <body>
3   <script language="JavaScript">
4   tableout = "<table border='1' width='300' cellspacing='0'
    cellpadding='3'>"
5   for (b = 1; b <= 10; b++) {
6   tableout = tableout + "<tr>":
7   for (g = 1; g <= 10; g++) {
8   tableout = tableout + "<td>" + b * g + "</td>";
9   }
10  tableout = tableout + "</tr>";
11  }
12  tableout = tableout + "</table>";
13  document.write (tableout);
14  </script>
15  </body>
16  </html>
```

(a) The code does not run as intended in web browsers because there is an error in the code.

Describe how the error in the code prevents the web browser from running as intended.

....

....

....

....

....

.... [2]

Question 1, Part (a)

Answer:

1(a)	**Two** *from:*	**2**
	A colon (:) is shown instead of a semi-colon (;) in line 6 ...this is a syntax error Syntax errors prevent JavaScript from being executed/are fatal errors in code The web browser displays nothing at all from this code The variable 'tableout' has not been declared before it is used Some browsers will ignore/compensate/interpret this differently from others Results can be different/unexpected in different browsers.	

Question 1, Part (b)

(b) Errors can be 'trapped' in order that the performance of the web browser is not affected.

Explain how you can use error handling techniques to trap errors in JavaScript code.

....

....

....

....

....

....

....

....

....

....

.... [5]

Question 1, Part (b)

Answer:

1(b)	**Five** *from:*	5
	Add specific code to deal with the errors transparently/without affecting the web browser Specify block of code to be tested Add some code produce output that depends on (the type of) error encountered Use try() block of code to be tested Use catch() to define the error handling code Use final() to allow code to be executed/run Use throw() to display information about the error/error message Specify the text to be displayed on screen as a result of the error.	

June 2020 – P31 & P33

8 Nigel is creating a website which uses JavaScript. He uses scripts to manipulate text held in HTML elements so that it can be displayed on his pages.

(a) Explain what is meant by an 'HTML element'.

....

....

....

....

....

.... [2]

Answer:

| 8(a) | *Two from:*
Component of a webpage/HTML document.
Surrounded/contained between tags.
Starting tag has <name of tag> and ending tag has </name of tag>
Node which can have attributes.
Node can have 'child nodes'.
Part of the Document Object Model (DOM) when browser has parsed/read/displayed the HTML into a page. | 2 |

Question 8, Part (b)

Nigel uses this code to manipulate the text on his page. The line numbers are shown only for reference purposes.

```
1
2   <html>
3   <body>
4
5   <h2>Nigel uses this JavaScript to change HTML</h2>
6
7   <p id="a1">Nigel's original text was here</p>
8
9   <script>
10  document.getElementById("a1").innerHTML = "Nigel changed this!";
11  </script>
12
13  <p>The paragraph above was changed by Nigel's script.</p>
14
15  </body>
16  </html>
```

(b) Explain, in detail, how this JavaScript would change what is displayed on the web page.

....

....

....

....

....

.... [2]

Answer:

8(b)	*Two from:* HTML document contains a <p> element with id="a1" HTML DOM is used to get the element with id="a1" (Line 7 changes the content of) innerHTML (to " Nigel changed this!").	2

June 2020 – P32

10 A teacher is creating a web page for his students to find out their result, Merit, Pass or Fail, by entering their test score. The teacher has chosen to use the JavaScript switch function to examine the test score entered by the student and report the result on the web page.

The code for the web page is shown in Fig. 10.1. The lines are numbered only for your convenience.

```
1
2   <html>
3   <body>
4
5   <input id="myTscore" type="number" value=0>
6   <button onclick="tscorelookup()">Check Your Result by entering your
    test score 0-100 </button>
7   <p id="myresult"></p>
8
9   <script>
10  function tscorelookup(){
11  var report;
12  var result = document.getElementById("myTscore").value;
13
14  switch(true){
15  case result < 0:
16  report = "You cannot score a mark below 0";
17  break;
18  case result > 100:
19  report = "You cannot score a mark above 100";
20  break;
21  case result >= 40:
22  report = "Your result is a Merit";
23  break;
24  case result >= 20:
25  report = "Your result is a Pass";
26  break;
27  case result < 20:
28  report = "Your result is a Fail";
29  break;
30  default:
31  report = "Please enter a valid mark";
32  }
33  document.getElementById("myresult").innerHTML = report;
34  }
35  </script>
36
37  </body>
38  </html>
```

Fig. 10.1

Question 10 (Continued)

The HTML code in lines 5 to 7 collects the test score entered by the student and, when the button is clicked, makes the test score available to the JavaScript code that starts at line 9.

Describe, in detail, how the JavaScript code works to handle a test score of 18.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [8]

Question 10

Answer:

| 10 | Eight from:
Line 9 <script> declares the code to be JavaScript.
Line 10 declares a function called tscorelookup()
Line 11 declares variable report.
Line 12 declares variable result.
Line 12 collects value/18 of 'myTScore' from user input into HTML code at line 5 and stores it in variable 'result'.
Line 14 'switch' function is used to compare the value in 'result' against pre-set 'case' values.
Line 15 checks value of 'result' to see if condition <0 is TRUE.
Line 18 checks value of 'result' to see if condition >100 is TRUE.
Line 21 checks value of 'result' to see if condition >=40 is TRUE.
Line 24 checks value of 'result' to see if condition >=20 is TRUE
…none of these are TRUE/all are untrue/all of these are FALSE
…control moves to next case.
Line 27 checks value of 'result to see if condition <20 is TRUE
…this is TRUE so control passes to Line 28 and FAIL comment is stored in variable 'report'.
Line 33 function displays contents of variable 'report' on webpage/displays "Your result is a Fail";
'break' is included to exit/jump out of any case
'default' is included in case no preceding case/condition is TRUE
Including 'default' is good coding practice even if (probably) not required. | 8 |

March 2020 – P32

9   In JavaScript, conditional statements can be used to carry out specific actions based upon different conditions. The value of the variable 'age' is entered into a web page and collected by the HTML code of the page. The function in Fig. 9.1 uses the 'if else' and 'else if' conditional statements to test the value of the variable 'age'. Line numbers are shown only for your convenience when referring to the code.

```
1   function myFunction() {
2     var statement;
3     var age = document.getElementById("age").value;
4   if (age < 10) {
5     statement = "You are not old enough to go to this school";
6   } else if (age < 18) {
7     statement = "You can go to this school";
8   } else {
9     statement = "You are too old to go to this school";
10  }
```

Fig. 9.1

(a)   Explain, with reference to the code in the function, how the 'if else' and 'else if' statements operate to provide the appropriate statement when an age is entered into the web page and collected by the HTML code.

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

Question 9, Part (a)

Answer:

9(a)	*Six from:* If statement specifies block of code that is executed if a condition is TRUE Line 4 if statement compares 'age' with condition <10 if TRUE 'You are not old enough…'is stored in variable 'statement' if FALSE execution is passed to line 6 Else-if statement specifies a new condition to be checked if the first condition is FALSE Line 6 else if statement compares 'age' with condition <18 if TRUE 'You can go to this school' is stored in variable 'statement' if FALSE execution is passed to line 8 and "You are too old to go to this school" is stored in variable 'statement' Else statement specifies the code to be executed if condition is FALSE.	6

Question 9, Part (b)

(b) The code could have been written using the 'switch' function, e.g. lines 4 to 7 could have been:

```
4    switch(true) {
5    case age < 10:
6      statement = "You are not old enough to go to this school";
7    break;
```

with additional lines for the other conditions.

Explain the drawbacks of using the 'switch' function.

....

....

....

....

....

....

....

....

....

.... [4]

Answer:

| 9(b) | *Four from:*
The order of Case/conditions checks in code must be logically correct/perfect
…for correct /expected comparisons to be made
(Switch) syntax does not follow the usual rules/colons not semi-colons so code is difficult/confusing to write/read
Code can be lengthy as each condition has to be individually stated
…this is repetitive and prone to error
'break' has to be manually inserted after every 'case'
….debugging problems/difficulties with 'nested' conditions
'default' condition should be included to catch/trap unexpected conditions. | **4** |