UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2018/2019

APRIL EXAMINATION

UECS2094/UECS2194 WEB APPLICATION DEVELOPMENT

FRIDAY, 3 MAY 2019 TIME : 2.30 PM – 4.30 PM (2 HOURS)

BACHELOR OF SCIENCE (HONS) APPLIED MATHEMATICS WITH COMPUTING
BACHELOR OF SCIENCE (HONS) SOFTWARE ENGINEERING

Instructions to Candidates :

This question paper consists of THREE (3) questions in Section A and TWO (2) questions in Section B.

Section A: Answer ALL questions.

Section B: Answer ONE (1) question only.

Each question carries 25 marks.

Note: Should a candidate answer more than ONE (1) question in Section B, marks will only be awarded for the FIRST question in Section B in the order the candidate submits the answer.

Answer the questions in the answer booklet provided.

An appendix containing selected HTML DOM and **XmlHttpRequest** properties and methods is provided.

Section A: Answer ALL questions.

Q1. (a) Describe the THREE (3) methods to add CSS styles to a website. Give ONE (1) example for each of the methods with the relevant HTML and/or CSS code. (6 marks)

(b) Write the HTML code to generate the following HTML table as shown in Figure 1.1. Include in the <head> section of the HTML file the relevant CSS style. (4 marks)

Student Name	Age	CGPA
Lim Goh Tong	18	3.50
Eve Hsia	20	3.00
John See	19	3.80

Figure 1.1

(c) Write the code to produce the HTML form as shown in Figure 1.2. Declare the required CSS styles as an embedded style sheet in your code. (15 marks)

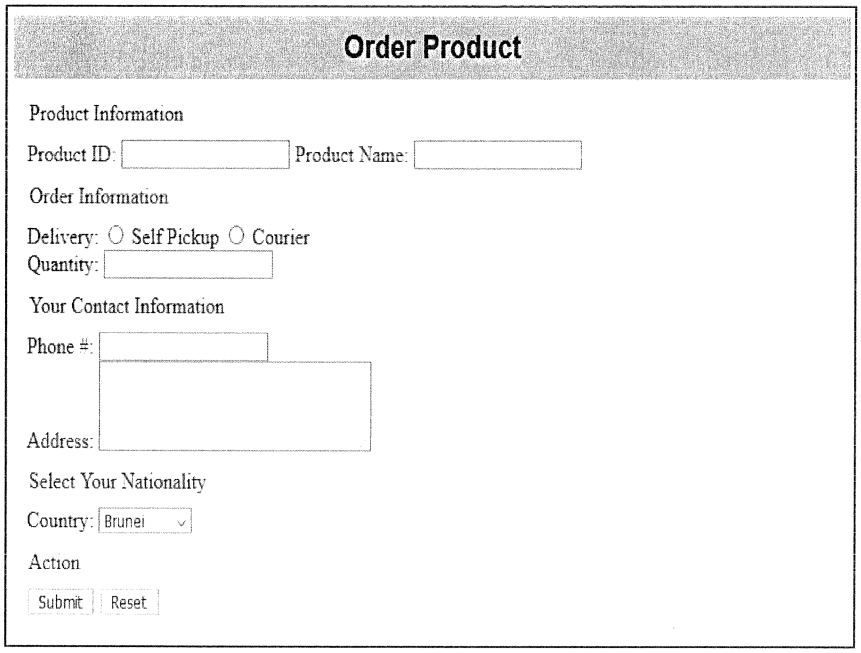

Figure 1.2

The following criteria for the form must be fulfilled:

- Use a fieldset to manage the display of the form elements.
- Fieldset padding is 10px.
- Texts are centered for the wording **Order Product**.
- Font family is **Arial** with a size of 25px for the wording **Order Product**.
- Font family is **Times** with a size of 18px for all form labels.
- The **<form>** element's action: "/post-order.php".
- The method for this form: **POST**

Q1(c). (Continued)

Table 1.3 is to be referred when naming and constructing the form elements.

Table 1.3

Form Element	Form element name	Data Type	Sample Data Value
Form			
Product ID	product-id	string	"PD1239100"
Product Name	product-name	string	"Badminton racket"
Quantity	quantity	int	10
Delivery	delivery	list (string)	"Self pickup", "Courier"
Phone #	phone	string	"011-3426456"
Address	address	string	"109, Jalan Tun Dato Seri, Kampung Cempaka, Petaling Jaya, Selangor."
Country	country	list (string)	"Brunei", "Indonesia", "Malaysia", "Singapore", "Vietnam"
Remarks	remarks	string	"Delivery within 1 week."

[Total : 25 marks]

Q2. (a) Write Javascript code to declare an object named **library** and initialize it with the values according to Table 2.1. (6 marks)

Table 2.1

title	author	quantity
The Road Ahead	Bill Gates	2
Steve Jobs	Walter Isaacson	5
Jungle Book	Collins Johnson	1

(b) Write the Javascript code to loop through the data structure you declared above and display the book title, author name and quantity in a HTML table as shown in Figure 2.2 (follow the CSS formatting given in Figure 2.3). (8 marks)

title	author	quantity
The Road Ahead	Bill Gates	2
Steve Jobs	Walter Isaacson	5
Jungle Book	Collins Johnson	1

Figure 2.2

Skeleton of the HTML code is shown below:

```
<html>
<head>
<title>Question 2(a)(b)</title>
<style>
table, th, td {
  border: 1px solid black;
}
table {
  border-collapse: collapse;
}
</style>
<script>
      //Q2(a). Declare library object here
</script>
</head>

<body>
<script>
      //Q2(b). write Javascript code to generate the HTML table
</script>
</body>
</html>
```

Figure 2.3

Q2. (Continued)

(c) Write the JavaScript "**removeFruit**" function to remove a fruit item selected from the list shown in Figure 2.4 below by clicking the button "**Select a fruit to remove**". (5 marks)

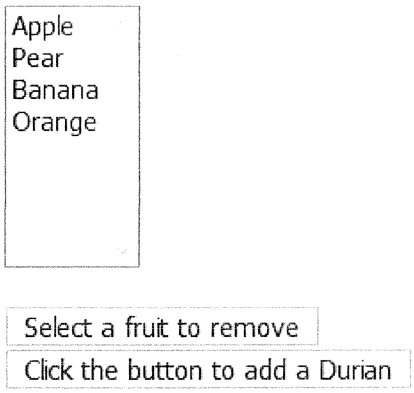

Figure 2.4

The HTML code skeleton that render the page shown in Figure 2.5 is shown as follows:

```
<html>
<head>
<style type="text/css">
  body {margin: 30px;}
</style>
<title>Add/Remove items from a dropdown list</title>
   <script>
   function removeFruit()
   {
       //Q2(c). Write your codes for this function
   }

   function insertFruit()
   {
       //Q2(d). Write your codes for this function
   }

</script>
</head>
<body>

<form>
<select id="fruitSelect" size="8">
     <option>Apple</option>
     <option>Pear</option>
     <option>Banana</option>
     <option>Orange</option>
</select>
```

Q2(c). (Continued)

```
</form><br>

<input type="button" onclick="removeFruit()" value="Select a
fruit to remove"><br>
<input type="button" onclick="insertFruit()" value="Click the
button to add a Durian">
</body>
</html>
```

Figure 2.5

(d) Using the same HTML code in part (c), write the JavaScript "**insertFruit**" function to insert an item named "**Durian**" to the list as shown in Figure 2.6 by clicking the button "**Click the button to add a Durian**". (6 marks)

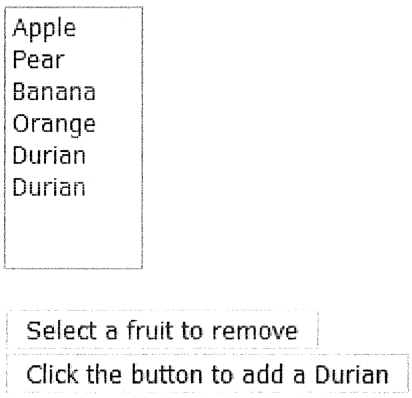

Figure 2.6

[Total : 25 marks]

Q3. (a) AJAX is a technique for creating fast and dynamic web pages. Example of applications that use AJAX are Google Maps, Facebook, and Gmail. Based on your understanding, draw a diagram to illustrate how AJAX works. (8 marks)

(b) Write code for page **demo.php** to implement POST request using AJAX. (2 marks)

(c) Figure 3.1 shows the rendering of a web-based image gallery. Initially, the gallery displays the first image from a set of images and its description as well as a scrollable list of thumbnails. Clicking on a thumbnail will replace the currently displayed image with the corresponding image and its description.

In this simplified implementation, the thumbnails are displayed using the actual image with their sizes specified in CSS.

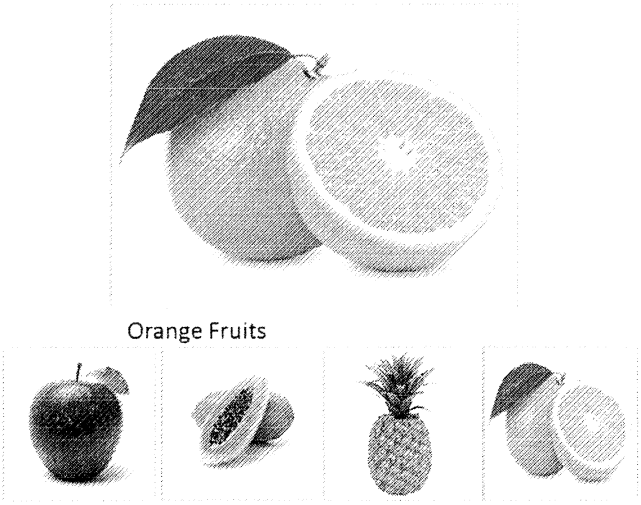

Figure 3.1

```
[{"file":"images\/apple.jpg","caption":"Apple
Fruits","alt":"Australian Apple Fruits"},{"file":"images\/
pineapple.jpg","caption":"Papaya Fruits ","alt":" Malaysian
Papaya Fruits "},{"file":"images\/ pineapple.jpg","caption":"
Pineapple Fruits ","alt":" Malaysian Pineapple Fruits
"},{"file":"images\/orange.jpg","caption":"Orange Fruits
","alt":" Australian Orange Fruits "}
```

Figure 3.2

Build the gallery shown in Figure 3.1 dynamically using the JSON data in Figure 3.2. The gallery in Figure 3.1 must have the document structure as shown in the HTML snippet in Figure 3.3.

Assume that the **div** elements with the **id** ***fruit_image*** and ***thumbnails*** have already been coded in plain HTML. You are NOT required to implement code for the styles. (15 marks)

Q3(c). (Continued)

```
<div id="fruit_image">
   <figure>
      <img src="images/orange.jpg" alt="Orange">
      <figcaption>
        Orange Fruit
      </figcaption>
   </figure>
</div>
<div id="thumbnails">
   <img src="images/apple.jpg" alt=" Australian Apple ">
   <img src="images/papaya.jpg" alt=" Malaysian Papaya Fruits">
   <img src="images/pineapple.jpg" alt="Malaysian Pineapple Fruits">
   <img src="images/orange.jpg" alt="Australian Orange Fruits">

</div>
```

Figure 3.3

[Total : 25 marks]

Section B: Answer ONE (1) question only.

Q4. (a) Write the complete statements in PHP to execute each of the following SQL operations. All the statements are related to each other.

(i) Create a new database called **employeeDB**. (2 marks)

(ii) Create a table named jobs including columns **job_id, job_title, min_salary, max_salary**. (5 marks)

(iii) Write the code to check whether the **max_salary** amount exceed the upper limit 25000 based on Q4(a)(ii)? (2 marks)

(iv) Write the statement for inserting the following data in Figure 4.1 in the **employee** table? (4 marks)

```
JOB_ID: 008
JOB_TITLE: Web Developer
MIN_SALARY: RM3000
MAX_SALARY: RM10000
```

Figure 4.1

(b) Figure 4.2 shows the output of the above mentioned in Figure 4.1 while Figure 4.3 shows the form with some specific actions by using PHP post method.

Output:

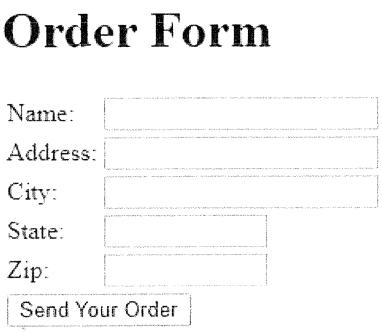

Figure 4.2

Q4(b). (Continued)

```
<html>

   <head>
        <title>Q4</title>
   </head>

   <body>

   (i)

 <h1>Order Form</h1>

<form method = "post" action = "/php/php_form.html">
 <table border="0">
<tr><td>Name:</td><td><input type="text" name="NAME" value=""></td></tr>
<tr><td>Address:</td><td><input type="text" name="ADDRESS"
value=""></td></tr>
<tr><td>City:</td><td><input type="text" name="CITY" value=""></TD></TR>
<tr><td>State:</td><td><input type="text" name="STATE" value=""
size="10"></td></tr>
<tr><td>Zip:</td><td><input type="text" name="ZIP" value=""
size="10"></td></tr>
<tr><td colspan="2"><input type="submit" value="Send Your
Order"></td></tr>

</table>
</form>
</body>
</html>
```

```
<?php
  echo "<h2>Your Given details are as :</h2>";

(ii)

?>
```

Figure 4.3

(i) Write the missing PHP - **post** method in box (i). (6 marks)

(ii) Write the missing PHP code in box (ii) to print the output after the user submits the order form. (6 marks)

[Total : 25 marks]

Q5. (a) Figure 5.1 shows the *var_dump* of the unserialized content of a cookie named **shoppingCart** with a few items added. Build a PHP script that reads the cookie named **shoppingCart** and produce the HTML table as shown in the example in Figure 5.2. You do NOT have to implement code for styles.

(9 marks)

```
array (size=4)
  0 =>
    array (size=4)
      'id' => string '33' (length=2)
      'name' => string 'London Tea set' (length=14)
      'quantity' => string '25' (length=2)
      'unit_price' => string '6' (length=1)
  1 =>
    array (size=4)
      'id' => string '19' (length=2)
      'name' => string 'London T-shirt' (length=14)
      'quantity' => string '2' (length=1)
      'unit_price' => string '99' (length=2)
  2 =>
    array (size=4)
      'id' => string '26' (length=2)
      'name' => string 'London Cap' (length=9)
      'quantity' => string '3' (length=1)
      'unit_price' => string '19' (length=2)
  3 =>
    array (size=4)
      'id' => string '20' (length=2)
      'name' => string 'London Fridge Magnet' (length=20)
      'quantity' => string '1' (length=1)
      'unit_price' => string '99' (length=2)
```

Figure 5.1

No	Item	Quantity	Unit Price	Total Price
1	London Tea set	1	25.00	25.00
2	London T-shirt	5	10.00	50.00
3	London Cap	3	8.00	24.00
4	London Fridge Magnet	10	5.00	50.00
			Grand Total	149.00

Figure 5.2

Q5. (Continued)

(b) (i) Write statements in PHP to execute SQL operations to create a table called ***user*** in a database, with the following fields (assume the specific database has already been accessed). (6 marks)

a. **userId**, which is the primary key and value is provided by the database server.
b. **name**, which has an appropriate string format.
c. **email**, with an appropriate string format.
d. **comment**, which has an appropriate string format.
e. **reg_date**, which has an appropriate date format.
f. **insert** one entry into the table.

(ii) A validation page has a form with fields for **name, email, comment** and **reg_date**. Write a ***validation.php*** script that checks if the **name, email** and **comment** are required fields. If the user submit the form without filling out the required fields, this function alerts a message, ***Please filled the required field**, and returns false, to prevent the form from being submitted. The users table is as described in Q5(b)(i). (10 marks)

[Total : 25 marks]

Appendix

Partial List of HTML DOM Methods

```
document.createElement(nodename)
document.createTextNode(text)
document.getElementById(elementID)
document.getElementsByName(name)
document.getElementsByTagName(tagname)
```

Partial List of HTML DOM Element Methods

```
node.appendChild(node)
node.removeChild(node)
node.replaceChild(newnode,oldnode)
element.setAttribute(attributename,attributevalue)
```

Partial List of HTML DOM Table Methods

```
tableObject.createCaption()
tableObject.createTHead()
tableObject.createTFoot()
tableObject.createTBody()
tableObject.insertRow(index)
theadObject.insertRow(index)
tfootObject.insertRow(index)
tbodyObject.insertRow(index)
trObject.insertCell(index)
```

Partial List of XMLHttpRequest Properties and Methods

```
object.open(method, url, [async], [user], [password])
object.send();
object.send(data);
object.onreadystatechange
object.readyState
object.responseText
```