USN										

Internal Assessment Test 2 – May 2023 scheme and solution

Sub:	WEBTECHNOLOGYANDITSAPPLICATIONS				Sub Code:	18CS63	Branch:	ISE	
Date:	24/05/2023	Duration:	90 min's	Max Marks: 50	Sem/Sec:	VI A, B & C		OBE	
Answer any FIVE FULL Questions							MARKS	CO	RBT

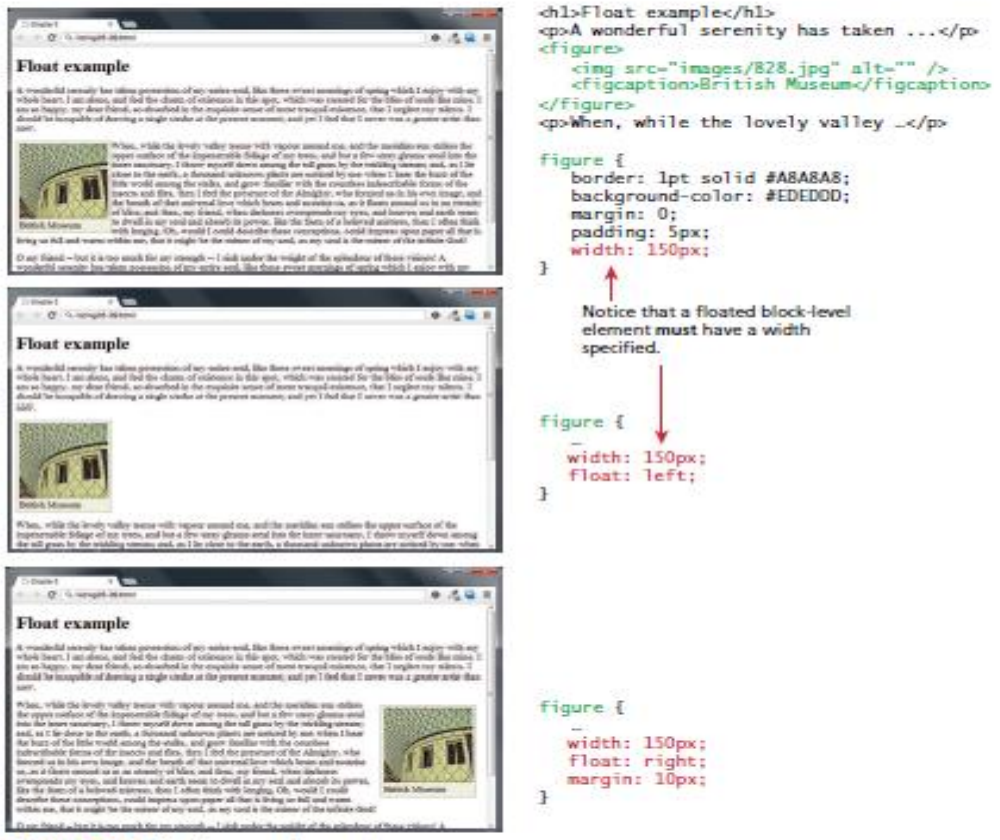
1. What does floating an element do in CSS? How do you float an element? 10 CO2 L2

Floating Elements:

- It is possible to displace an element out of its position in the normal flow via the CSS float **property**.
- An element can be floated to the left or floated to the right.
- When an item is floated, it is moved all the way to the far left or far right of its containing block and the rest of the content is "re-flowed" around the floated element, as can be seen in Figure 5.9.
- Notice that a floated block-level element must have a width specified; if you do not, then the width will be set to auto, which will mean it implicitly fills the entire width of the containing block, and there thus will be no room available to flow content around the floated item. Also note in the final example in Figure 5.9 that the margins on the floated element are respected by the content that surrounds the floated element.

Floating within a Container:

- It should be reiterated that a floated item moves to the left or right of its container (also called its **containing block**). In Figure 5.9, the containing block is the HTML document itself so the figure moves to the left or right of the browser window.



```
<h1>Float example</h1>
<p>A wonderful serenity has taken ...</p>
<figure>
   <img src="images/828.jpg" alt="" />
   <figcaption>British Museum</figcaption>
</figure>
<p>When, while the lovely valley …</p>

figure {
   border: 1pt solid #A8A8A8;
   background-color: #EDEDDD;
   margin: 0;
   padding: 5px;
   width: 150px;
}
```

Notice that a floated block-level element must have a width specified.

```
figure {
   …
   width: 150px;
   float: left;
}

figure {
   …
   width: 150px;
   float: right;
   margin: 10px;
}
```

FIGURE 5.9 Floating an element

2. What is responsive design? Why is it important? Explain in detail. 10 CO2 L2

In a responsive design, the page “responds” to changes in the browser size that go beyond the width scaling of a liquid layout. One of the problems of a liquid layout is that images and horizontal navigation elements tend to take up a fixed size, and when the browser window shrinks to the size of a mobile browser, liquid layouts can become unusable.

In a responsive layout, images will be scaled down and navigation elements will be replaced as the browser shrinks There are now several books devoted to responsive design, so this chapter can only provide a very brief overview of how it works.

There are four key components that make responsive design work.
They are:
1. Liquid layouts
2. Scaling images to the viewport size
3. Setting viewports via the tag
4. Customizing the CSS for different viewports using media queries Responsive designs begin with a liquid layout, that is, one in which most elements have their widths specified as percentages.

Making images scale in size is actually quite straightforward, in that you simply need to specify the following rule:

```
img { max-width: 100%; }
```

Of course this does not change the downloaded size of the image; it only shrinks or expands its visual display to fit the size of the browser window, never expanding beyond its actual dimensions. More sophisticated responsive designs will serve different sized images based on the viewport size.

3. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
a) Parameter: A string
b) Output: The position in the string of the left-most vowel
c) Parameter: A number
d) Output: The number with its digits in the reverse order 10 CO3 L2

```
<!DOCTYPE HTML>

<html>
  <body>
      <script type="text/javascript">
      var str = prompt("Enter the Input","");

      if(!(isNaN(str)))
      {
            var num,rev=0,remainder;
            num = parseInt(str);
            while(num!=0) {
                  remainder = num%10;
                  num = parseInt(num/10);
                  rev = rev * 10 + remainder;
```

```
                    }
                    alert("Reverse of "+str+" is "+rev);
            }
            else
            {
                    str = str.toUpperCase();
                    for(var i = 0; i < str.length; i++) {
                            var chr = str.charAt(i);
                            if(chr == 'A' || chr == 'E' || chr == 'I' || chr == 'O' || chr == 'U')break;
                    }
                    if( i < str.length )
                            alert("The position of the left most vowel is "+(i+1));
                    else
                            alert("No vowel found in the entered string");
         }
            </script>

     </body>
    </html>
```

4. Explain 3 forms of linking Javascript to HTML page with suitable code segments. 10 CO3 L2

1. Inline JavaScript

Inline JavaScript refers to the practice of including JavaScript code directly within certain HTML attributes, such as that shown in Listing 1.1.

listing 1.1 Inline JavaScript example

```
<a href="JavaScript:OpenWindow();"more info</a>
<input type="button" onclick="alert('Are you sure?');" />
```

listing 1.2 Embedded JavaScript example

```
<script type="text/javascript">
/* A JavaScript Comment */
alert ("Hello World!");
</script>
```

You may recall that in Chapter 3 on CSS you were warned that inline CSS is in general a bad practice and should be avoided. The same is true with JavaScript. In fact, inline JavaScript is much worse than inline CSS. Inline JavaScript is a real maintenance nightmare, requiring maintainers to scan through almost every line of HTML looking for your inline JavaScript.

2. Embedded JavaScript

Embedded JavaScript refers to the practice of placing JavaScript code within a <script> element as shown in Listing 1.2. Like its equivalent in CSS, embedded JavaScript is okay for quick testing and for learning scenarios, but is frowned upon for normal realworld pages. Like with inline JavaScript, embedded scripts can be difficult to maintain.

3. External JavaScript

Since writing code is a different competency than designing HTML and CSS, it is often advantageous to separate the two into different files. JavaScript supports this separation by allowing links to an external file that contains the JavaScript, as shown in Listing 1.3.
This is the recommended way of including JavaScript scripts in your HTML pages.
By convention, JavaScript external files have the extension .js. Modern websites often have links to several, maybe even dozens, of external JavaScript files (also called **libraries**). These external files typically contain function definitions, data definitions, and other blocks of JavaScript code.

listing 1.3 External JavaScript example

```
<head>
<script type="text/JavaScript" src="greeting.js">
</script>
</head>
```

5. Discuss Arrays of Javascript in detail. 10 CO3 L2

Arrays are one of the most used data structures. In practice, this class is defined to behave more like a linked list in that it can be resized dynamically, but the implementation is browser specific, meaning the efficiency of insert and delete operations is unknown.

The following code creates a new, empty array named greetings:

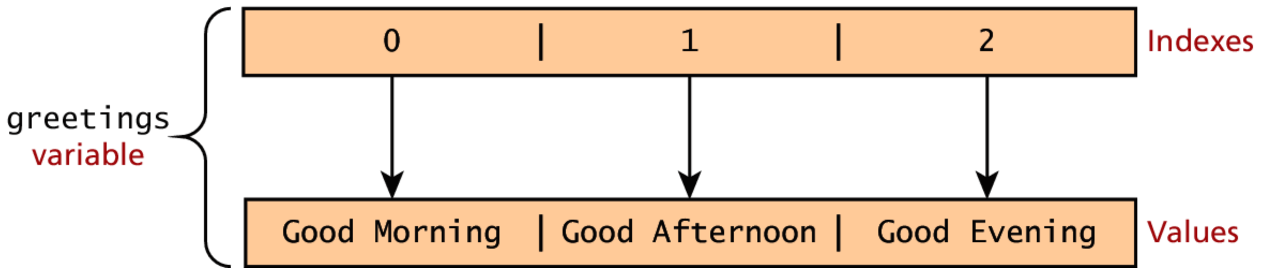
```
var greetings = new Array();
```

To initialize the array with values, the variable declaration would look like the following:

```
var greetings = new Array("Good Morning", "Good Afternoon");
```

or, using the square bracket notation:

```
var greetings = ["Good Morning", "Good Afternoon"];
```



To add an item to an existing array, you can use the **push** method.

greetings.**push**("Good Evening");

The **pop** method can be used to remove an item from the back of an array.

Additional methods: concat(), slice(), join(), reverse(), shift(), and sort()

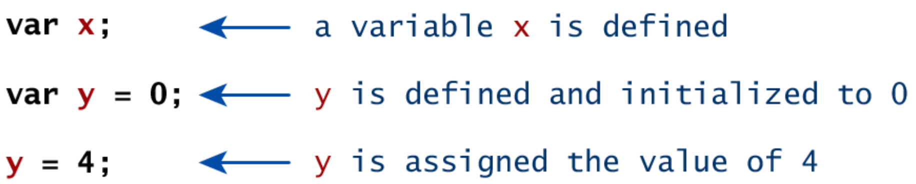
6. Write the PHP programs to do the following:

I. Describe PHP data types.

PHP supports the following data types:

- String

- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL



```
var x;       ←—— a variable x is defined

var y = 0;   ←—— y is defined and initialized to 0

y = 4;       ←—— y is assigned the value of 4
```

II. Multiplication of two matrices.

III. Addition of two matrices. 10 CO4 L3

```
<?php
        $a = array(array(1,2,3),array(4,5,6),array(7,8,9));
        $b = array(array(7,8,9),array(4,5,6),array(1,2,3));
        $m=count($a);
        $n=count($a[2]);
        $p=count($b);
        $q=count($b[2]);
        echo "the first matrix  :"."<br/>";
        for ($row = 0; $row < $m; $row++) {
                for ($col = 0; $col < $n; $col++)
                     echo " ".$a[$row][$col];
                echo "<br/>";
        }
        echo "the second matrix        :"."<br/>";
        for ($row = 0; $row < $p; $row++) {
            for ($col = 0; $col < $q; $col++)
                    echo " ".$b[$row][$col];
                echo "<br/>";
        }
        echo "the transpose for the first matrix
        is:"."<br/>"; for ($row = 0; $row < $m;
        $row++) {
            for ($col = 0; $col < $n; $col++)
                   echo " ".$a[$col][$row];
                echo "<br/>";
        }
if(($m===$p) and ($n===$q)) {

                echo "the addition of matrices is:"."<br/>";
                for ($row = 0; $row < 3; $row++) {
                        for ($col = 0; $col < 3; $col++)
                        echo " ".$a[$row][$col]+$b[$row][$col]." ";
                        echo "<br/>";
                }
        }
        if($n===$p){
                echo " The multiplication of matrices: <br/>";
                $result=array();
                for ($i=0; $i < $m; $i++) {
                        for($j=0; $j < $q; $j++){
                                $result[$i][$j] = 0;
                                for($k=0; $k < $n; $k++)
                                        $result[$i][$j] += $a[$i][$k] * $b[$k][$j];
                        }
                }
                for ($row = 0; $row < $m; $row++) {
                        for ($col = 0; $col < $q; $col++)
                                echo " ".$result[$row][$col];
                        echo "<br/>";
                }
        }
?>
```