In MVC C#, what is the role of _ViewStart.cshtml?

Selected
Answer:
B.
It is used to specify common settings for all the views
under a folder and sub-folders where it is created.

Question 4

In the following example, the two divs have the same height and width
but render with different sizes. How can you ensure that they appear
the same size (height 100px and width 300px) when rendered?

```
<style>
.div1 {
  width: 300px;
  height: 100px;
  border: 1px solid blue;
}
.div2 {
  width: 300px;
  height: 100px;
  padding: 50px;
  border: 1px solid red;
}
</style>
```

Selected Answer: B. Set box-sizing: border-box to both divs

Question 5

Consider the following HTML code:
```
<html>
 <head>
  <link type="text/css" rel="stylesheet" href="stylesheet.css" />
  <title>HTML Reference</title>
  <style>
    div > p, .active {
    color:blue;
    }
  </style>
 </head>
 <body>
  <p>Introduction: Cascading with CSS</p>
     <div>
        <p>First paragraph
        </p>
        <ul>
          <li><p class="active">First item</p></li>
          <li><p>Second item </p></li>
```

```
            <li><p>Third item</p></li>
        </ul>
        <p>Last paragraph</p>
    </div>
  </body>
</html>
```

Which of the text will appear in blue when the file is rendered?

Selected Answer: First paragraph, Last paragraph
                    D.

Question 6

What is the output of the JavaScript as provided below?

```
function getItem(arr) {

    const number = Math.floor(Math.random() * arr.length);
    const item = arr[number];
    return item;
}
const array = [2, 'ready', 15, 80];
const result = getItem(array);
console.log(result);
```

The output is

Selected   D.
Answer: A random choice of one of the values 2, ready, 15 or 80

Question 7

Consider the code fragment below:

```
var b = 7;
console.log(b);
{
    var a = 30;
    console.log(b+5);
}
console.log(a);
let arr = ['work', 'play', 'eat'];
console.log(arr);
```

```
arr[2] = 'hello';
console.log(arr);
```

The output of the above code is:

Selected Answer: 7
                              12
                              30
                              ['work', 'play', 'eat']
                              ['work', 'play', 'hello']

                  A.

Question 8

Consider the following code snippet.

```
var allBagElements = [];
// setter

localStorage.setItem("allBagElements", JSON.stringify(allBagElements));
//missing code for the getter
```

What is the missing code?

Selected   D.
Answer: var allBagElements =
                  JSON.parse(localStorage.getItem("allBagElements"));

Question 9

Which of the following is NOT true of ViewBag in ASP.NET MVC?

Selected Answer: D. It transfers data from the view to the controller.

Question 10

Consider the HTML table below.

```
<table>
 <tr>
  <td>Tyron</td>
  <td>Lewin</td>
  <td onclick="delete_row(this)">Lindiwe</td>
 </tr>
</table>
```

What will be the effect after the cell is clicked and the JavaScript code below is executed?

```
function delete_row(e) {
        e.parentElement.parentElement.remove();
    }
```

Selected Answer: The table will be deleted

                D.

Question 11

This script listens for **keyup** events in an input field, retrieves the value entered, calculates the length of the string, and displays the length in a designated element. When the document is ready, the script attaches a **keyup** event listener to the element with the ID "my_addr". Upon releasing a key in this input field, the script performs the following actions:

1. It retrieves the value entered in the input field with the ID "my_addr".
2. It calculates the length of the retrieved string.
3. It sets the HTML content of the element with the ID "addr_length" to the calculated length of the string.

Select the correct option below:

Selected Answer: $(document).ready(function() {
                         $('#my_addr').keyup(function(){
                         var my_str2=$('#my_addr').val();
                         $('#addr_length').html(my_str2.length)
                         })
                         })
                    C.

Question 12

This script captures the click event on an element, retrieves the value entered in an input field, calculates the length of the string, and displays the length on a designated element. When the document is ready, the script attaches an event listener to the element with the ID "my_name". Upon clicking this element, the script performs the following actions:

1. It retrieves the value entered in the input field with the ID "my_input".
2. It calculates the length of the retrieved string.
3. It sets the HTML content of the element with the ID "my_display" to the calculated length of the string

Select the correct option below:

Selected Answer:

```
$(document).ready(function() {
$("#my_name").click(function(){
var my_str=$('# my_input").val();
$('#my_display').html(my_str.length)
})
})
```

B.

Question 13

If you run the code fragment below, which explanation best describes what will happen

```
$(document).ready(function(){
$("#p1").hover(function(){
alert("You entered p1!");
 },
function(){
alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph</p>
```

Select the correct option below:

Selected   C.
Answer:
                1. When the document is ready
                2. The element with ID = p1 is selected
                3. The sentence "This is a paragraph" which is p1 is
                   displayed
                4. When the mouse hovers over the paragraph, a
                   message "You entered p1!" is displayed
                5. When the hovers over the paragraph a second time,
                   a message "You entered p1!" is displayed using the
                   alert function.

Question 14

This script provides a search functionality for a table. It filters and
displays the table rows based on the entered search value, and
highlights the matching rows by adding a "highlight" class. When a
keyup event occurs on the element with the ID "search", the script
performs the following actions:

1. It retrieves the value entered in the "search" input field and
   converts it to lowercase.
2. It filters the table rows within the tbody of the element with the
   ID "myTable".
3. For each filtered table row, it retrieves the text content and
   converts it to lowercase.
4. It checks if the entered search value is present in the row text
   using the indexOf() method.
5. It toggles the visibility of the table row based on whether it
   matches the search value.
6. If the row matches the search value, it adds the "highlight" class
   to highlight the row.
7. If the row doesn't match the search value, it removes the
   "highlight" class to remove any previous highlighting.

Select the correct option below:

Selected Answer:

```
$("#search").on("keyup", function () {
var value = $(this).val().toLowerCase();
$("#myTable tbody tr").filter(function () {
var rowText = $(this).text().toLowerCase();
var matches = rowText.indexOf(value) > -1;
$(this).toggle(matches);
if (matches) {
$(this).addClass("highlight");
} else {
$(this).removeClass("highlight");
}
});
});
```

C.

Question 15

This script adds a new row to a table when the "addButton" is clicked, using the values entered in the input fields for name, surname, and email. The indicated script adds data to an HTML table upon clicking a button. When the "addButton" element is clicked, the script performs the following actions:

1. It retrieves the trimmed values of the "name," "surname," and "email" input fields.
2. It checks if all three values (name, surname, and email) are not empty or false.
3. If all three values are provided, it constructs an HTML string representing a table row with the retrieved values.
4. It appends the constructed table row to the tbody element of a table with the ID "myTable."
5. It clears the values of the "name," "surname," and "email" input fields.
6. It disables the "addButton" element.

Selected Answer:

```
$("#addButton").on("click", function () {
var name = $("#name").val().trim();
var surname = $("#surname").val().trim();
var email = $("#email").val().trim();
if (name && surname && email) {
var row = "<tr><td>"
+ name + "</td><td>"
+ surname + "</td><td>"
+ email + "</td></tr>";
$("#myTable tbody").append(row);
$("#name, #surname, #email").val("");
$("#addButton").prop("disabled", true);
}
});
```

D.

Question 16

This script handles an update action when the "btnUpdate" button is clicked. It retrieves input values, updates specific table cells in the "tblCustomers" table, shows the "btnAdd" button, hides the "btnUpdate" button, and calls the "Clear()" function. When the element with the ID "btnUpdate" is clicked, the script performs the following actions:

1. It initializes variables for name, country, and id.
2. It retrieves the values entered in the input fields with the IDs "txtId", "txtName", and "txtCountry" respectively.
3. It selects the table rows within the tbody of the element with the ID "tblCustomers".
4. It finds the specific table row based on the value stored in the element with the ID "hfRowIndex".
5. It selects the second and third table cells (td) within the found row.
6. It sets the HTML content of the second table cell to the retrieved name value.
7. It sets the HTML content of the third table cell to the retrieved country value.
8. It shows the element with the ID "btnAdd".
9. It hides the element with the ID "btnUpdate".
10. It calls the "Clear()" function.

Select the correct option below:

Selected   B.
Answer:

```
$('#btnUpdate').on('click', function () {
var name, country, id;
id = $("#txtId").val();
name = $("#txtName").val();
country = $("#txtCountry").val();
$('#tblCustomers tbody
tr').eq($('#hfRowIndex').val()).find('td').eq(1).html(name);
$('#tblCustomers tbody
tr').eq($('#hfRowIndex').val()).find('td').eq(2).html(country
)
$('#btnAdd').show();
$('#btnUpdate').hide();
Clear();
});
```

Question 17

Consider the following HTML and JavaScript (JS) code

INDEX.HTML

```html
<html lang="en">
    <head>
     <meta charset="UTF-8" />
      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
      <title>INF272Test</title>

    </head>
    <body>
     <main>
       <div class="container">
        <h1>Test</h1>
        <span id="value">0</span>
        <div class="button-container">
          <button class="btn one">One</button>
          <button class="btn two">Two</button>
          <button class="btn three">Three</button>
        </div>
       </div>
     </main>
     <!-- javascript -->
     <script src="app.js"></script>
    </body>
   </html>
```

APP.JS

```
// set initial value to zero
```

```
let x = 0;
// select value and buttons
const value = document.querySelector("#value");
const btns = document.querySelectorAll(".btn");
btns.forEach(function (btn) {
  btn.addEventListener("click", function (e) {
    const styles = e.currentTarget.classList;
    if (styles.contains("one")) {
      x--;
    } else if (styles.contains("three")) {
      x++;
    } else {
      x = 0;
    }
    if (x > 0) {
      value.style.color = "green";
    }
    if (x < 0) {
      value.style.color = "red";
    }
    if (x === 0) {
      value.style.color = "blue";
    }
    value.textContent = x;
  });
});
```

In a few lines (briefly) indicate what the result will be when the code is executed.

Selected Answer: [None Given]

Question 18

Review the code provided, and then briefly indicate, step-by-step, the general functionality of the script when the window being rendered loads. Make use of an ordered list to organize the sequence of your answer's steps.

```
<script type="text/javascript">
  window.onload = function () {
    document
      .getElementById("submit")
      .addEventListener("click", function () {
        let date = new Date();
        let table = document.getElementById("tbody");
        table.innerHTML
          += `<tr>`
          + `<td class="tableSimpleStyle">${date.toLocaleTimeString()}</td>`
          + `<td class="tableSimpleStyle"><a class='delete' href='JavaScript:void(0);'><button type='button'
                                    class='button'>Remove</button></a></td>`
```

```
                    + `</tr>`;
            });

        $("#table").on("click", ".delete", function () {
            $(this).closest('tr').remove();
        });

        $("button").click(function () {
            var rowCount = $("#table tr").length - 2;
            alert("There are " + rowCount + " additional rows.");
        });

    }
</script>
```

Selected Answer: [None Given]

Question 19

Review the jQuery script provided below and describe the code behaviour. Give a general description of the functionality of the script, and specific sections (1 & 2). Make use of an ordered list to organize the sequence of your answer's steps.

Section 01

```
<script>
    $(document).ready(function () {
        $("#name, #surname, #email").on("keyup", function () {
            var name = $("#name").val().trim();
            var surname = $("#surname").val().trim();
            var email = $("#email").val().trim();
            if (name && surname && email) {
                $("#addButton").prop("disabled", false);
            } else {
                $("#addButton").prop("disabled", true);
            }
        });
```

Section 02

```
        $("#addButton").on("click", function () {
            var name = $("#name").val().trim();
            var surname = $("#surname").val().trim();
            var email = $("#email").val().trim();
            if (name && surname && email) {
                var row = "<tr><td>"
                    + name + "</td><td>"
                    + surname + "</td><td>"
                    + email + "</td></tr>";
                $("#myTable tbody").append(row);
```

```
            $("#name, #surname, #email").val("");
            $("#addButton").prop("disabled", true);
        }
    });
});
```

i. Generally describe the behaviour of the entire script (2 marks).

ii. Describe the behaviour of the script in Section 01 (4 marks).

iii. Describe the behaviour of the script in Section 02 (4 marks).

Selected Answer: [None Given]

Question 20

Consider the HTML file below:

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>INF272Test</title>
    <style>
        table, th, td {
            border: 1px solid black;
        }
    </style>
  </head>
  <body>
    <main>
      <div class="container">
        <table id="body">
          <tr>
            <th>Column one</th>
            <th>Column two</th>
          <th>Column three</th>
          </tr>
          <tbody >
```

```
    <tr>
     <td>(0,0)</td>
     <td>(0,1)</td>

    </tr>
    <tr>
     <td>(1,0)</td>
     <td>(1,1)</td>
      <td>(1,2)</td>

    </tr>
   </tbody>

  </table>
 </div>
</main>
<script src="app.js"></script>
</body>
</html>
```

Provide the code in app.js such that the table has three rows as indicated below:

Column one	Column two	Column three
(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2.0)	(2,1)	(2,2)

Selected Answer: let table = document.getElementById("Body");
console.log(table);
let row = document.createElement("tr");
let c1 =document.createElement("td");
let c2 =document.createElement("td");
let c3= document.createElement("td");
c1.innerText ="(2,0)";
c2.innerText ="(2,1)";
c3.innerText ="(2,2)";
row.appendChild(c1);
row.appendChild(c1);
console.log(row);
table.appendChild(row);

Question 21

Consider the need to create a record for a typical student of INF 272 with the following attributes: *student number, Firstname, Surname, Age, Address, Grade*.

Show the code to create an MVC Model from this. Make sure the properties have the correct data types.

Selected Answer: namespace List.Models
```
{
public class Student
 { public int studentnumber {get ;set}
   public string  Firstname {get ;set}
   public string Surname {get ;set}
  public  int  Age {get ;set}
    public string  Address {get ;set}
    public int Grade {get ;set}
 }
 }
```

Monday, 30 December 2024 14:16:37 o'clock SAST

← **OK**