CMRIT
CELEBRATING 25 YEARS
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – May 2022

Sub:	WEB TECHNOLOGY AND ITS APPLICATIONS			Sub Code:	18CS63	Branch:	ISE		
Date:	09/05/2022	Duration:	90 min's	Max Marks:	50	Sem/Sec:	VI A, B & C		OBE
Answer any FIVE FULL Questions							MARKS	CO	RBT
1	Design the given form in table layout and use appropriate styles as shown in figure.						10	CO2	L3
2	What are the three cascade principles used by browsers when style rules conflicts? Briefly describe each.						10	CO1	L1
3 (a)	Explain the role of the following commands in HTML5 with syntax i) <nav> ii) <figure> iii) <aside>						6	CO1	L1
3 (b)	Compare id selector and class selector in CSS with suitable example.						4	CO1	L2
4(a)	What is CSS? Illustrate how it is added to HTML in different ways with suitable examples						6	C01	L2
4(b)	. Explain the role of and HTML tags with example.						4	CO1	L1
5	What are contextual selectors? Explain the four different contextual selectors.						10	CO 2	L2
6	Explain and describe the difference between a relative and an absolute reference. When should each be used?						10	CO2	L1

Faculty Signature   CCI Signature   HOD Signature

1. **Design the given form in table layout and use appropriate styles as shown in figure.**

Form within Table

Name	[ ]
Password	[ ]
Email	[ ]
Contact no.	[ ]
Country	Choose a country ▾
Permant Address	enter address with pincode
Gender	○ Male ○ Female
Languages known	☐ C ☐ Java ☐ Python ☐ C++
	Submit Reset

Solution:

<html>

<head>

 <style>

  table,th{

   border:solid 1pt black;

   border-collapse: collapse;

   padding: 15px;

  }

  td{

   border:solid 1pt black;

   padding:10px;

  }

```css
    tr:nth-child(odd)
    {
      background-color: cornsilk;
    }
    tr:nth-child(even)
    {
      background-color:darksalmon
    }
    th{
      background-color: cyan;
    }
    .merged{
        text-align: center;
    }
  </style>
</head>
<body>
<h2>Form within Table</h2>
<form method="GET" action=" "></form>
<table border = "border">
    <tr>
      <td>
         Name
      </td>
      <td>
         <input type="text" name="name"/>
      </td>
    </tr>
     <tr>
       <td>
          Password
       </td>
       <td>
          <input type="password" name="password"/>
```

```
          </td>
        </tr>
        <tr>
          <td>
            Email
          </td>
          <td>
            <input type="email" name="email">
          </td>
        </tr>
        <tr>
          <td>
            Contact no.
          </td>
          <td>
            <input type="tel" name="contactno">
          </td>
        </tr>
        <tr>
          <td>
            Country
          </td>
          <td>
            <select name="where">
              <option>Choose a country</option>
              <option>India</option>
              <option>Australia</option>
              <option>United States</option>
            </select>
          </td>
        </tr>
        <tr>
          <td>
            Permant Address
```

```html
      </td>
      <td>
        <textarea placeholder="enter address with pincode"></textarea>
      </td>
    </tr>
    <tr>
      <td>
        Gender
      </td>
      <td>
        <input type="radio" name="gender" value="male">Male
        <br>
        <input type="radio" name="gender" value="female">Female
      </td>
    </tr>
    <tr>
      <td>
        Languages known
      </td>
      <td>
        <input type="checkbox" name="C"/>
        <label>C</label> <br>
        <input type="checkbox" name="Java"/>
        <label>Java</label> <br>
        <input type="checkbox" name="Python"/>
        <label>Python</label> <br>
        <input type="checkbox" name="PHP"/>
        <label>C++</label>
      </td>
    </tr>

    <tr>
      <td class="merged" colspan="2">
        <input type="submit"/>
```

```html
            <input type="reset"/>
        </td>
    </tr>
</table>


</body>
</html>
```

2. **What are the three cascade principles used by browsers when style rules conflicts? Briefly describe each.**

Solution :

The Cascade: How Styles Interact

Multiple CSS rules can be defined for the same HTML element, at different locations – inline, embedded or external. The browser determines the style to be applied on an element, depending on the location and hierarchy of the html element.

The "Cascade" in CSS refers to how conflicting rules are handled. CSS uses the following cascade principles to help it deal with conflicts: inheritance, specificity, and location.

Inheritance

Inheritance is the first of these cascading principles. Many (but not all) CSS properties affect not only themselves but their descendants as well. Font, color, list, and text properties are inheritable; layout, sizing, border, background, and spacing properties are not inheritable.

If suppose, this is a document,
```
<head>
<style>
    body {
    font-family: Arial;
    color: red;
    border: 8pt solid green;
    margin: 100px;
    }

    div {
    font-weight: bold;
    }
</style>
</head>
<body>
<div>Will be displayed in red, with arial font and bold</div>
</body>
```

The font settings are inherited from the parent tag, border and margin are not inheritable. However it is possible to tell elements to inherit properties that are normally not inheritable, by explicitly specifying as 'inherit'.

```
    div {
    font-weight: bold;
    border: inherit;
    margin: inherit;
    }
```

Specificity

Specificity is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element. In CSS, the more specific the selector, the more it takes precedence (i.e., overrides the previous definition)

```
<head>
<style>
    body {
    font-family: Arial;
    color: red;
    border: 8pt solid green;
    margin: 100px;
    }

    div {
    font-weight: bold;
    color: blue;
    }
</style>
</head>
<body>
<div>Will be displayed in blue, with arial font and bold</div>
</body>
```

The content of <div> is displayed in blue, as the red color setting of <body> tag is overridden in the specification of <div> tag.

Location

The principle of location is that when rules have the same specificity, then the latest are given more weight. Ie., an inline style will override one defined in an embedded style sheet and embedded style will override the external style sheet.

Styles defined in external style sheet X will override styles in external style sheet Y if X's <link> element is after Y's in the HTML document.
<link rel= "stylesheet" href= "Y">
<link rel= "stylesheet" href= "X">

When the same style property is defined multiple times within a single declaration block, the last one will take precedence.

Specificity algorithm:
■ First count 1 if the declaration is from a "style" attribute in the HTML, 0 otherwise (let thatvalue = a).
■ Count the number of ID attributes in the selector (let that value = b).
■ Count the number of class selectors, attribute selectors, and pseudo-classes in the selector (letthat value = c).
■ Count the number of element names and pseudo-elements in the selector (let that value = d).
■ Finally, concatenate the four numbers a+b+c+d together to calculate the selector's specificity.

3. Explain the role of the following seman_c elements of HTML5 with syntax and script segments
        i) <nav>
        ii) <figure>
        iii) <aside>

Navigation

The <nav> element represents a section of a page that contains links to other pages or to other parts within the same page. Like the other new HTML5 semantic elements, the browser does not apply any special presentation to the <nav> element. The <nav> element was intended to be used for major navigation blocks. However, like all the new HTML5 semantic elements, from the browser's perspective, there is no definite right or wrong way to use the <nav> element. Its sole purpose is to make the document easier to understand.

```html
<header>
   <img src="logo.gif" alt="logo" />
   <h1>Fundamentals of Web Development</h1>
   <nav role="navigation">
     <ul>
         <li><a href="index.html">Home</a></li>
         <li><a href="about.html">About Us</a></li>
         <li><a href="browse.html">Browse</a></li>
     </ul>
   </nav>
</header>
```

igure and Figure Captions

Prior to HTML5, web authors typically wrapped images and their related captions within a nonsemantic <div> element. In HTML5 we can instead use the <figure> and <figcaption> elements. *The figure element represents some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the main flow of the document.*

```html
<p>This photo was taken on October 22, 2011 with a Canon EOS 30D camera.</p>
<figure>
     <img src="images/central-park.jpg" alt="Central Park" /><br/>
     <figcaption>Conservatory Pond in Central Park</figcaption>
</figure>
</p>
```

The above tags illustrates a sample usage of the <figure> and <figcaption> element.

Aside

The <aside> element is similar to the <figure> element, the <aside> element "represents a section of a page that consists of content that is indirectly related to the content around the aside element". The <aside> element is be used for sidebars, pull quotes, groups of advertising images, or any other grouping of non-essential elements.

(b) Compare id selector and class selector in CSS with suitable example.

Solution: <u>Class Selectors</u>
A **class selector** allows to simultaneously target different HTML elements. The HTML elements with the same class attribute value, can be styled by using a class selector.
Syntax: period (.)classname{ styles}

Eg:
```
<head>
    <title>Student details </title>
    <style>
    .first {
    Font-style: italic;
    Color:red;
    }
    </style>
</head>
<body>
    <h1 class="first">Student Info</h1>
    <div>
    <p class="first">Amith</p>

    <p>Easy to learn.</p>
    </div>
    <hr/>
    <div>
    <p class="first">Bhushan</p>
    <p>Very much special.</p>
  </div>
  <hr/>
</body>
```

<u>Id Selectors</u>
An **id selector** allows to assign style to a specific element by its id attribute.
Syntax: hash (#)id name

Eg:
```
<head>
    <title>Student details </title>
    <style>
    #first{
    Font-style: italic;
    Color:red;
    }
    </style>
</head>
<body>
    <h1 id="first">Student Info</h1>
    <div>
```

```
            <p id="first">Amith</p>
            <p>Easy to learn.</p>
            </div>
            <hr/>
            <div>
            <p >Bhushan</p>
            <p>Very much special.</p>
            </div>
            <hr/>
      </body>
```

4. **(a)** Explain the role of <ul> and <ol> HTML tags with example.
    Solution:

Lists

HTML provides simple and effective ways to specify lists in documents.
There are three types of lists:

■ **Unordered lists**. Collections of items in no particular order; these are by default rendered by the browser as a bulleted list. However, it is common in CSS to style unordered lists without the bullets. Unordered lists have become the conventional way to markup navigational menus.

■ **Ordered lists**. Collections of items that have a set order; these are by default rendered by the browser as a numbered list.


```
<html >
```

```
<head> <title> list </title>
</head>
<body>
    <h3> Cessna 210 Engine Starting Instructions </h3>
    <ul >
        <li>sssss,</li>

        <li>saaaaa</li>
    </ul>
    <hr size="5"  />
    <ol >
        <li> Set propeller to high RPM </li>
        <li> Set ignition switch to "BOTH" </li>
        <li> Set auxiliary fuel pump switch to "LOW PRIME" </li>
        <li> When fuel pressure reaches 2 to 2.5 PSI, push starter button </li>
        </ol>
        </body>
        </html>
```
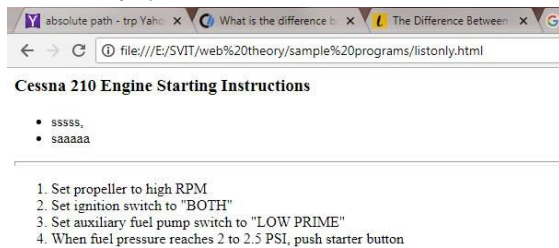

4 (b) What is CSS? Illustrate how it is added to HTML in different ways with suitable examples.

Solution : Cascading style sheets

Inline Styles
Inline styles are style rules placed within an HTML element using the style attribute, as shown below. An inline style only affects the element it is defined within and overrides any other style definitions. Selector is not necessary with inline styles and that semicolons are only required for separating multiple rules.

Disadvantages of using inline style-
Style is applied to an element only
Maintaining the inline style is difficult

The advantage of using inline style is that it can be quickly tested for a style change.

Eg: <h2 style = "font-size:24pt;"> Description</h2>
       <h2 style = "font-size:24pt; font-weight:bold;"> Reviews </h2>

Embedded Style Sheet (Document Level/Internal)
Embedded style sheets (also called **internal styles or document level styles**) are style rules placed within the <style> element (inside the <head> element of an HTML document) and apply to the whole body of the document.

The disadvantage of using embedded styles is that it is difficult to consistently style multiple documents when using embedded styles. But it is helpful when quickly testing out a style that is used in multiple places within a single HTML document. Spaces are ignored in <style> element.

```
<head>
<title>Student Data</title>
<style>
h1 { font-size: 24pt; }
h2 {
font-size: 18pt;
font-weight: bold;
}
</style>
</head>
<body>
<h1>Student count</h1>
<h2>CSE/ISE Department</h2>
……..
</body>
```

External Style Sheet
External style sheets are style rules placed within an external text file with the **.css** extension. This style provides the best maintainability. When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version.

To reference an external style sheet, you must use a <link> element (within the <head> element). Several style sheets can be linked at a same time. Each linked style sheet will require its own <link> element.

```
<head >
<title>Share Your Travels -- New York - Central Park</title>
<link rel="stylesheet" href="styles.css" />
</head>
```

5. What are contextual selectors? Explain the four different contextual selectors.

Contextual Selectors

A **contextual selector** (in CSS3 also called **combinators**) allows to select elements based on their *ancestors*, *descendants*, or *siblings*. It selects elements based on their context or relation to other elements in the document tree.

As shown in below table, **descendant selector** matches all elements that are contained within another element. The character used to indicate descendant selection is the space character.

Selector	Matches	Example
Descendant	A specified element that is contained somewhere within another specified element.	div p Selects a <p> element that is contained some-where within a <div> element. That is, the <p> can be any descendant, not just a child.
Child	A specified element that is a direct child of the specified element.	div>h2 Selects an <h2> element that is a child of a <div> element.
Adjacent sibling	A specified element that is the next sibling (i.e., comes directly after) of the specified element.	h3+p Selects the first <p> after any <h3>.
General sibling	A specified element that shares the same parent as the specified element.	h3~p Selects all the <p> elements that share the same parent as the <h3>.

```
<head>
    <title>Student details </title>
    <style>
    #first p{
    font-style: italic;
    color: red;
    }
    </style>
</head>
<body>
    <h1 id="first">Student Info
    <div>
    <p >Amith</p>
    <p>Easy to learn.</p>
    </div>
    </h1>
    <hr/>
    <div>
    <p >Bhushan</p>
```

Output:
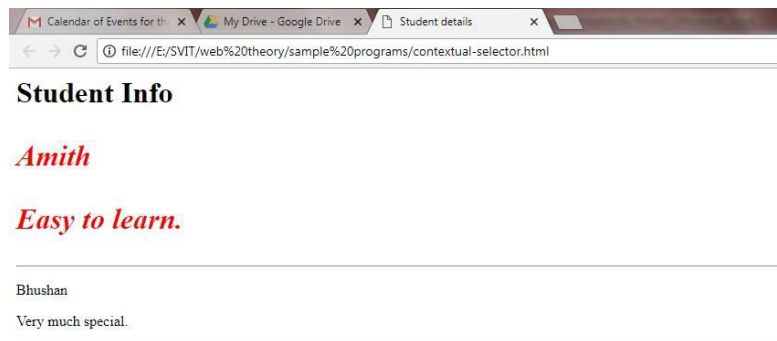
```
    <p>Very much special.</p>
    </div>
    <hr/>
</body>
```

6. Explain and describe the difference between a relative and an absolute reference. When should each be used?

URL Relative Referencing

- To construct links with the <a> element, reference images with the <img> element, or include external JavaScript or CSS files, the files should be successfully referred using **relative referencing** from the document.
- If the referred file is an external file then **absolute reference** is required. Absolute path contains the full path including the domain name, any paths, and then finally the file name of the desired resource.
- However, when referencing a resource that is on the same server as the HTML document, then **relative referencing** is done. Relative paths change depending upon where the links are present.

There are several rules to create a link using the relative path:

- links in the same directory as the current page have no path information listed **filename**
- sub-directories are listed without any preceding slashes **weekly/filename**
- links up one directory are listed as **../filename**

Sl. No.	Relative Link Type	Example
1	**Same Directory** – To link to a file within the same folder, simply use the file name.	<a href = "ex.html">
2	**Child Directory** – To link to a file within a subdirectory, use the name of subdirectory and a slash before the file name.	<a href = "images/ex.html">
3	**Grandchild/Descendant Directory** – To link to a file that is multiple subdirectories below the current one, construct the full path by including each subdirectory name before the file name.	<a href = "css/images/ex.html">
4	**Parent/Ancestor Directory** – use "../" to reference a folder above the current one. If trying to reference a file several levels above the current one, simply string together multiple "../".	<a href = "../ex.html"> <a href = "../../ex.html">
5	**Sibling Directory** – use "../" to move up to the appropriate level, and then use the same technique as for child or grandchild directories.	<a href = "../ex.html"> <a href = "../images/ex.html">
6	**Root Reference** – An alternative approach for ancestor and sibling references is to use the root reference approach. Ie. Begin the reference with the root reference ("/") and then use the same technique as for child or grandchild directories.	<a href = "/ex.html"> <a href = "/images/ex.html">
7	**Default Document** -Web servers allow references to directory names without file names. In such a case, the web server will serve the default document.	<a href = "members"> or <a href = "/members">