Roll No.: _

Amrita Vishwa Vidyapeetham

Amrita School of Engineering, Coimbatore

B.Tech missed Assessment Examinations – October 2022

Seventh Semester

19CSE461-NET CENTRIC PROGRAMMING

Missed Periodical Exam(Solution Key)

Duration: Two hours   Maximum: 50 Marks

Course Outcomes (COs):

CO	Course Outcomes
1	**Apply the concepts of responsive web design to customize pages for users need**
2	**Apply markup and scripting languages to design and validate dynamic pages**
3	**Evaluate the appropriateness of client server applications using mean stack architecture**
4	**Design, develop and deploy client/server applications**
5	Implement the complete web application development cycle as a Term project

1. Find the difference between the given two code samples. Identify the output of each.

```
a) myData = {"name":"John", "age":30, "car":null};
b) Data = '{"name":"John", "age":30, "car":null}';
   myData = JSON.parse(Data);
```

Ans:

a) Creation of a JavaScript object from a JSON object literal
b) Creation of a JavaScript object by parsing a JSON string

Both will produce the output as ‘Object’ data type.

2. Find the value of ‘text’ in the following java script:

```
const Data = '{"name":"John", "age":30, "car":null}';
const myData = JSON.parse(Data);

let text = "";
for (const x in myData) {
  text += x + ", ";
}
```

Ans:

text = name, age, car,

3. Write a node-js script to display the contents (input data) of url.

Ans:

```
var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(8080);
```

For example, if your url contains month and year as shown below, then this will display its respective value given in the url.

http://localhost:8080/?year=2017&month=July

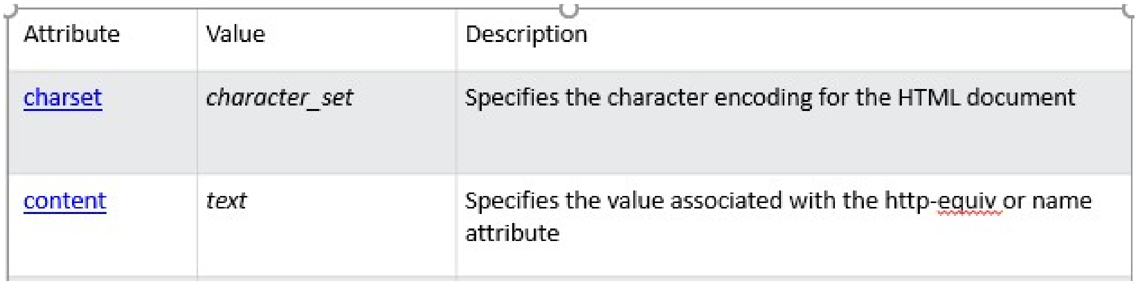
Parsing an address with the url.parse() method, and it will return a URL object with each part of the address as properties

4. List the various node-js file system methods available for accessing the files.

Ans:

fs.readFile()
fs.appendFile()
fs.writeFile()
fs.open()
fs.unlink()
fs.rename()

5.



Attribute	Value	Description
charset	*character_set*	Specifies the character encoding for the HTML document
content	*text*	Specifies the value associated with the http-equiv or name attribute

6.
The HTML DOM defines a standard way for accessing and manipulating HTML documents. It presents an HTML document as a tree-structure.

The XML DOM defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure.

7.

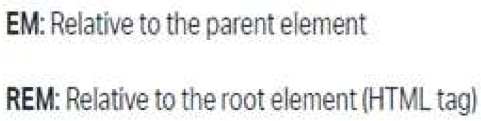
```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

8.Give the meaning of following notations in relative View Port length

a. EM

b.REM



EM: Relative to the parent element

REM: Relative to the root element (HTML tag)

9.

```
<div ng-app="" ng-init="myCol='lightblue'">
<input style="background-color:{{myCol}}" ng-model="myCol">
</div>
```

10.

```
<div ng-app="" ng-nit="person={firstName:'John',lastName:'Doe'}">
<p>The name is {{ person.firstName }}</p>
<p>The name is {{ person.lastName }}</p>
</div>
```

Part-B 5*6 =30

11. Write a Node-js script to create your own module to add two numbers and use the module to add the given 2 numbers.

MyModule.js

```
exports.add = function (x,y) {
        return x + y;
};
```

AddNumbers.js

```
var http = require('http');
var dt = require('./MyModule');

http.createServer(function (req, res) {
 res.writeHead(200, {'Content-Type': 'text/html'});
 res.write("Addition: " + dt.add(6,6));
 res.end();
}).listen(8080);
```

12.

Routing refers to determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on).

Each route can have one or more handler functions, which are executed when the route is matched.

Route definition takes the following structure:

```
app.METHOD(PATH, HANDLER)
```

where:

- app is an instance of express.
- METHOD is an HTTP request method, in lowercase.
- PATH is a path on the server.
- HANDLER is the function executed when the route is matched.

Example

Respond with `Hello World!` on the homepage:

```
app.get('/', (req, res) => {
  res.send('Hello World!')
})
```

13.

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
    </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
    <xsl:value-of select="description"/>
    <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per serving)</span>
    </p>
  </div>
</xsl:for-each>
```

```
</body>
</html>
```

14.

```
{
 "catalog": {
   "book": {
     "author": "Gambardella, Matthew",
     "title": "XML Developer's Guide",
     "genre": "Computer",
     "price": 44.95,
     "publish_date": "2000-10-01",
     "description": "An in-depth look at creating applications \n      with XML."
   }
 },
 "book": {
   "author": "Galos, Mike",
   "title": "Visual Studio 7: A Comprehensive Guide",
   "genre": "Computer",
   "price": 49.95,
   "publish_date": "2001-04-16",
   "description": "Microsoft Visual Studio 7 is explored in depth,\n      looking at how Visual Basic, Visual C++, C#, and ASP+ are \n      integrated into a comprehensive development \n      environment."
 }
}
```

15.

Angular JS .js

```
<html>
   <head>
     <title>Angular JS Includes</title>
     <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
     </script>
   <style>
      table, th , td {
         border: 1px solid grey;
         border-collapse: collapse;
         padding: 5px;
      }
      table tr:nth-child(odd) {
         background-color: #f2f2f2;
      }
      table tr:nth-child(even) {
         background-color: #ffffff;
      }
   </style>
</head>

<body>
   <h2>AngularJS Sample Application</h2>

   <div ng-app = "mainApp" ng-controller = "studentController">
      <div ng-include = "'/angularjs/src/include/main.htm'"></div>
      <div ng-include = "'/angularjs/src/include/subjects.htm'"></div>
   </div>

   <script>
      var mainApp = angular.module("mainApp", []);

      mainApp.controller('studentController', function($scope) {
         $scope.student = {
            firstName: "Mahesh",
            lastName: "Parashar",
            fees:500,

            subjects:[
              {name:'Physics',marks:70},
              {name:'Chemistry',marks:80},
              {name:'Math',marks:65},
              {name:'English',marks:75},
              {name:'Hindi',marks:67}
            ],
            fullName: function() {
              var studentObject;
              studentObject = $scope.student;
              return studentObject.firstName + " " + studentObject.lastName;
            }
         };
      });
   </script>

</body>
</html>
```

<u>Personaldetails.htm</u>

```
<table border = "0">
   <tr>
      <td>Enter first name:</td>
      <td><input type = "text" ng-model = "student.firstName"></td>
   </tr>

   <tr>
      <td>Enter last name: </td>
      <td><input type = "text" ng-model = "student.lastName"></td>
   </tr>

   <tr>
      <td>Name: </td>
      <td>{{student.fullName()}}</td>
   </tr>
```

```
</table>
```

Subjects.htm

```
<p>Subjects:</p>
<table>
  <tr>
    <th>Name</th>
    <th>Marks</th>
  </tr>

  <tr ng-repeat = "subject in student.subjects">
    <td>{{ subject.name }}</td>
    <td>{{ subject.marks }}</td>
  </tr>
</table>
```