Name______________________________

Web Application Programming

(CS472)

January 2022

Final Examination

SCI Question (5 points)

Write one main point (1 - 2 sentences) on any aspect of *Node.js*, along with a corresponding SCI point. Do not use any of the main points from the Main Point Charts in the slides. (Use some other SCI point than "Do Less and Accomplish More.")

Instructions: For each Multiple Choice Question, select one answer.

Questions on jQuery

1. When used in a jQuery event handler function $(this) refers to:
(a) the window object
(b) the event handler function
(c) the body of the event handler
(d) the object containing the instruction
[✓ (e)] the HTML element originating the event

2. Which jQuery method returns the direct parent element of the selected element?
[✓ (a) parent( )]
(b) parents( )
(c) ancestor( )
(d) ancestors( )
(e) above( )

3. What is the correct jQuery code to set the background color of all p elements to red?
(a) $("p").manipulate("background-color","red");
[✓ (b) $("p").css("background-color","red");]
(c) $("p").layout("background-color","red");
(d) $("p").style("background-color","red");
(e) $("p").background("red");

4. With jQuery, look at the following selector: $("div.intro"). What does it select?
(a) The first div element with class="intro"
[✓ (b) All div elements with class="intro"]
(c) All div elements with id="intro"
(d) The first div element with id="intro"
(e) Not a valid selector

5. Which jQuery method is used to set one or more style properties for selected elements?
(a) html()
[✓ (b) css()]
(c) style()
(d) view()
(e) visual()

6. Which jQuery function is used to prevent code from running, before the document is finished loading?
(a) $(body).onload( )
[✓ (b) $(document).ready( )]
(c) $(document).load( )
(d) window.onload( )
(e) all of the above

7. Look at the following selector: $("div p"). What does it select?
[✓ (a) All p elements inside a div element]
(b) The first p element inside a div element
(c) All div elements with a p element
(d) Not a valid selector
(e) All div elements

8. Which jQuery method is used to remove selected elements?
(a) remove()
(b) detach()
[✓ (c) Both methods can be used]

9.
```
function logger() {
  return function () {
    return function () {
      console.log('Message');
    }
  }
}
```
Which of the following logs 'Message' to the console?
[✓ (a) logger( );]
(b) logger( ) ( );
(c) logger( ) ( ) ( );
(d) logger((( )));
(e) Console.log(logger( ));

10. Which of the following jQuery method sets the html contents of an element?

[✓ (a) html(val)]

(b) setHtml(val)

(c) setInnerHtml(val)

(d) innerHTML(val)

(e) None of the above

11. Which of the following jQuery method inserts the specified content at the end of the selected elements.

(a) $(selector).add(content)

(b) $(selector).addElement(content)

(c) next(selector).add(content)

[✓ (d) $(selector).append(content)]

(e) None of the above

12. Which of the following jQuery methods gets a set of elements containing the unique next siblings of each of the given set of elements?

(a) locateNext(selector)

(b) getNext(selector)

[✓ (c) next(selector)]

(d) after(selector)

(e) None of the above

---

13. The purpose of the following HTML web page and Javascript code is to display a list of items, each with a *Delete* button. When the User clicks on a *Delete* Button, the corresponding list item (with its Button) disappears from the list. Your job is to write the code of the Javascript *DeleteItem*( ) function. This Javascript will be running in the Browser. Write your answer on the answer sheet.

<u>HTML:</u>

```
<h1>Shopping Cart</h1>
<ul>
   <li> Shirt  <button>Delete</button>  </li>
   <li> Tie    <button>Delete</button>  </li>
   <li> Jacket <button>Delete</button>  </li>
   <li> Hat    <button>Delete</button>  </li>
</ul>
```

<u>Javascript:</u>

```
(function( ) {
  function DeleteItem( ) {
     ...  Insert your code here
  }
  $(document).ready( function( ) {
     $("button").click(DeleteItem);
  });
})();
```

14. Write a jQuery instruction to remove the first paragraph from the body of an HTML document.

Questions on Node.js

Instructions: For each Multiple Choice Question, select one answer.

15. Suppose you want to count visits to your website? Which of the following would be the best technique?
(a) Permanent Cookie on the server
(b) Temporary Cookie on browser
[✓ (c)] Permanent Cookie on browser
(d) local variable in your web server code
(e) global variable in express

16. When using the cookie-parser module, which of the following will retrieve the value of the "username" cookie from the current HTTP request?
[✓ (a)] req.cookies.username
(b) request.getCookies('username')
(c) Cookie c = request.getCookie('username')
(d) res.cookie('username');
(e) res.redirect.cookie.username

17. Suppose you are writing an online shopping web application that has a login page. Once a user enters password information, we would like to give the user the option of saving that information, so they do not have to enter it the next time they visit the website from the same machine even if it is days or weeks later. Which of the following would be the best technique?
(a) object in the session
(b) array in the session
(c) Temporary Cookie on the server
[✓ (d)] Permanent Cookie on browser
(e) Temporary Cookie on browser

Consider the following Node.js Web Server.

```
const app = require('express')();
app.use((req, res, next) => {
  console.log(`${req.url}`);
  next();
});
app.get('/test', (req, res, next) => {
  res.send('ok');
});
var server = app.listen(5000, function () {
    console.log('Server is running..');
});
```

18. For the following URL entered in the Browser, what will appear on the Server Console and Browser window:
    http://localhost:5000/test

(a) Console: ok Browser: ok
[✓ (b) Console: /test Browser: ok]
(c) Console: /test Browser: /test
(d) Console: GET Browser: ok
(e) Console: 404 Browser: Error

Consider the following Pug file (*index.pug*) in the \views folder:

```
doctype 5
html
  body
    h1 Hello, world!
    p= message
```

and the following Web Server:

```
const express = require('express');
const app = express();
app.set('views', __dirname + '/views');
app.set('view engine', 'pug');
app.get('/', function (request, response) {
    response.render('index', { message: 'I enjoy life' });
  });
var server = app.listen(5000, function () {
    console.log('Server is running..');
});
```

19. What will appear in the Browser Window in response to the URL http://localhost:5000/
[✓ (a) Hello World! I enjoy life]
(b) Hello World! Server is running..
(c) p = Hello World!
(d) I enjoy life I enjoy life
(e) Hello World! message

Consider a Web Site consisting of the following two files:

index.html

```
<!DOCTYPE html>
   <html>
      <head><meta charset="utf-8"/>
         <title>Test</title>
      </head>
      <body>
          <form name="form" method="post" action="http://localhost:8080/submit-data">
              <input type="text" name="name" value="OK"/>
              <input name="submit" type="submit" value="OK"/>
          </form>
      </body>
   </html>
```

Server.js

```
var express = require('express');
var app = express();
const path = require('path');
app.use(express.urlencoded({ extended: false }));
app.get('/', function (req, res) {
    res.sendFile(path.join(__dirname, 'index.html'));
});
app.post('/submit-data', function (req, res) {
    res.send("name:  " + req.body.name + " ||| OK:  " + req.body.OK);
});
var server = app.listen(8080, function () {
    console.log('Node server is running..');
});
```

20. The User enters the following URL in the Web Browser: `http://localhost:8080/` and then submits the form without making any changes. What is the resultant output?
(a) name: OK ||| OK: OK
(b) name: undefined ||| OK: OK
[✓ (c)] name: OK ||| OK: undefined
(d) name: OK ||| OK: 8080

---

21. A user types the URL http://www.withoutbook.com/Blog . Which HTTP request gets generated?
[✓ (a)] GET method
(b) POST method
(c) HEAD method
(d) PUT method

22. When using HTML forms which of the following is true for the POST method?
(a) POST allows users to bookmark URLs with parameters.
(b) The POST method should not be used when large amount of data needs to be transferred.
(c) POST uses the FTP internet protocol.
[✓ (d)] POST method sends data in the body of the request.

Consider the following Node.js Web Server.

```
const express = require('express');
const app = express();
app.use(function (request, response, next) {
  response.writeHead(200, { 'Content-Type': 'text/plain' });
  next();
});
app.get('/', function (request, response) {
  response.end('Welcome to the homepage!');
});
app.get('/about', function (request, response) {
  response.end('Welcome to the about page!');
});
app.use(function (request, response) {
  response.statusCode = 404;
  response.end('Page not found:  404 Error');
});
var server = app.listen(5000, function () {
    console.log('Server is running..');
});
```

For each of the following URLs entered in the Browser, which message will appear in the Browser window:

23. URL: http://localhost:5000/

[✓ (a)] `Welcome to the homepage!`
(b) `Welcome to the about page!`
(c) `Page not found:  404 Error`
(d) `Server is running..`

24. URL: http://localhost:5000/about

(a) `Welcome to the homepage!`
[✓ (b)] `Welcome to the about page!`
(c) `Page not found:  404 Error`
(d) `Server is running..`

25. URL: http://localhost:5000/trees

(a) `Welcome to the homepage!`
(b) `Welcome to the about page!`
[✓ (c)] `Page not found:  404 Error`
(d) `Server is running..`

26. URL: http://localhost:5000/about_trees

(a) `Welcome to the homepage!`
(b) `Welcome to the about page!`
[✓ (c)] `Page not found:  404 Error`
(d) `Server is running..`

Consider the following Node.js Web Server.

```
const express = require('express');
const app = express();
app.post('/data', function (req, res) {
    res.send('Wiki data is posted');
app.get('/', function(req, res) {
  res.send('Wiki home');
});
app.get('/docs', function(req, res) {
  res.send('Documentation for this wiki');
});
var server = app.listen(5000, function () {
    console.log('Server is running..');
});
```

27. What will appear in the Browser Window in response to the URL http://localhost:5000/docs
(a) Server is running..
(b) Wiki home
(c) Wiki data is posted
(d) data::docs
[✓ (e)] Docmentation for this wiki

Assume MySQL is running on the *localhost* and has a database called *entries*, with a table also called *entries* having three fields: *word*, *wordtype*, *definition*. Consider the following portion of Node.js code executing on the *localhost*:

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "myPassword",
  database: "entries"
});
var meaning;
con.connect(function(err) {
  if (err) throw err;
  con.query(`SELECT definition FROM entries WHERE word = "tree" `,
            function (err, result) {
                  if (err) throw err;
                  meaning = result;
            });
});
console.log(meaning);
```

28. The console output of this code is: *undefined*. Which of the following will correct this error and allow the definition of the word "tree" to appear on the console?
(a) increase the clock speed of the computer processor
(b) replace MySQL with Microsoft SQL Server
[✓ (c)] move `console.log(meaning)` to the end of the `con.query` callback function
(d) create and use a new User Name and Login for MySQL
(e) clear the queue of requests to MySQL before executing the code

29. Consider a web site *Multiply* whose opening page contains two text boxes and a submit button. The User enters a number into each text box and then clicks the submit button. A new page then appears in the Browser window with the product of the two numbers. A *node.js* Web Server to implement this *Multiply* web site is as follows:

```
const express = require("express");
const app = express();

app.use(express.urlencoded( { "extended": false}));

app.get("/", (req, res) => {
  res.send(`<form action='http://localhost:3000/multiply' method='POST'>
             <input text name='num1'>
             <input text name='num2'>
             <input type='submit' value='Multiply Numbers'>
           </form>`);
});

/* insert another route here to compute product and send HTML to Browser */

app.listen(3000);
```

Your job is to complete this Web Server by inserting another route as indicated in the comment. Write your answer on the answer sheet.

```
app.post('/multiply',function(req,res){
   const multi=(req.body.num1*req.body.num2);

 res.send(`<!DOCTYPE html>
<html>
<head><meta charset=\"utf-8\"/>
<title>Multiply</title>
</head>
<body>
<p>The Answer is: ${String(multi)}</p>
</body>
</html> `);
```

True/False Questions

Instructions: Indicate *True* or *False* for each statement.

F 1. When using the Pug Template Engine, whitespace and indentations are ignored.

T 2. Javascript code or expressions may appear in a Pug Template.

T 3. Node.js is an open-source server side runtime environment built on Chrome's V8 JavaScript engine.

T 4. Node.js provides an event driven, asynchronous runtime environment for building server-side applications using JavaScript.

T 5. Node.js uses coverage to delegate events to different runtime threads.

F 6. Whenever a new event arrives, the Node.js runtime will delete all pending events from the queue.

T 7. With Node.js, all the user requests to your web application will be handled by a single thread.

T 8. Node.js is not fit for an application which performs CPU-intensive operations like image processing or other heavy computation work.

F 9. The Node.js runtime allows a maximum of three middleware modules in a web server.

T 10. In Node.js, an event loop is constantly watching for the events to be raised for an asynchronous job and executing a callback function when the job completes.

F 11. A Javascript Promise object returns a jQuery AJAX request.

T 12. The client-side Javascript instruction **`x = $("p")`** stores a jQuery object in variable **`x`**.

F 13. HTML Form data always goes directly to the Server, it cannot be accessed by Javascript running on the Browser.

F 14. An XMLHttpRequest to a Web Server must always return data in XML format.

T 15. When using a jQuery $.ajax( ) call, webpage form data can be sent to the server.

T 16. When using jQuery $.ajax( ), both Get and Post requests are allowed.

F 17. By using jQuery $.ajax( ), the Javascript code running in the Web Browser can access SQL Databases stored on the client computer.

F 18. If a Javascript function has *n* parameters, then any call of the function can have at most 2**n* arguments.