COMP 2406 Fall 2018 Midterm Test -ANSWERS

Scantron Test: Submit answers on the Scantron card provided, you can keep the test paper. Questions each worth 1 mark (20 marks total).

1) Given the following typed into a browser address field: http://localhost:3000/canvasWithTimer.html
Which ONE of the following is FALSE about what the URL means?

A) Accessing http://localhost:3000 uses the TCP/IP stack.
B) canvasWithTimer.html may or may not be an actual html file on the server.
C) Localhost refers to the computer the browser is running on.
D) The port 3000 specifies that the request will use the http POST method.
E) Localhost is the portion of the URL for which we could substitute an IP address.

2) Which ONE of the following statements about the HTTP protocol is FALSE?

A) HTTP is the transport protocol that world-wide-web applications use.
B) Both HTTP GET and POST can transport user data.
C) HTTP can use ports other than 3000.
D) HTTP is stateless (has no notion of previous messages).
E) HTTP GET is more useful than POST because it can transport arbitrary length user data.

3) Which ONE of the following is TRUE regarding the difference between a GET and POST HTTP request?

A) GET can communicate user data but POST cannot.
B) POST requests are encrypted but GET requests are not.
C) GET query parameters are visible in the browser address.
D) GET HTTP requests can be inspected by routers but POST requests cannot.
E) POST data is limited in length but GET query parameters are not.

4) JSON is a popular format for exchanging data between client and server. Which ONE of the following statements about JSON is FALSE?

A) JSON is less verbose than its competitor XML.
B) JSON data are not objects but rather string representations of objects.
C) An NPM helper module is required to parse JSON data in javascript running in Node.js.
D) JSON can be transported with a MIME type 'application/json' or 'text/plain'.
E) JSON data can be transported with an AJAX GET or POST request.

5) Which ONE of the following is FALSE about JSON encoding?

A) JSON strings must use double quotes to describe object property names.
B) Not all javascript objects can be converted to JSON.
C) JSON can describe data but not javascript functions.
D) JSON can contain date objects (i.e. dates don't have to be turned into strings).
E) JSON string notation is similar to javacript literal object notation.

6) Given two javascript objects X and Y which ONE of the following best describes the difference between javascript expressions:

```
X == Y   //line 1
X === Y  //line 2
```

A) Line 1 is generally considered safer than line 2.
B) Line 1 is used when X and Y are javascript objects but line 2 is used when X and Y are JSON strings.
C) Line 2 is only available in javascript ES6 but line one is available in the older ES5.
D) Line 2 compares arguments without type casting or type conversion. Line 1 allows type conversion.
E) Line 2 is not legal because === should only be used for comparisons between strings.

7) Which ONE of the following is **FALSE** regarding the difference among variables declared with `var`, `let`, or `const` in Javascript?

A) Variables declared with `var` are hoisted to function scope but those declared with `let` are not.
B) Uninitialized `let` variables can be referenced before they are declared in later statements.
C) Variables declared with `var` have function scope but those declared with `let` have block scope.
D) Variables declared with `let` can re-assigned but those declared with `const` cannot.
E) Variables declared without `var`, `let`, or `const` are legal and have global scope.

8) Given the following two loops in javascript (which differ only in the use of `var` vs. `let`) which ONE of the following best describes the difference, if any?

```
for(var i = 0; i<10; i++){  //LOOP 1
  setTimeout(function() {console.log(i);}, 1000)
}

for(let i = 0; i<10; i++){  //LOOP 2
   setTimeout(function() {console.log(i);}, 1000)
}
```

A) Both loops will output the same result.
B) LOOP 1 will print 10,10,10, … whereas LOOP 2 will print 0,1,2, …
C) LOOP 1 will print 0,1,2 … whereas LOOP 2 will print 10,10,10, …
D) LOOP 1 will start printing before the for loop ends.
E) LOOP 2 will start printing before the for loop ends.

9) Consider the following Javascript code:

```
function Car(){
   this.set = function(k,v){this[k]=v}
}

let c = new Car()
let f = c.set      //line 1
f('engine','v8')   //line 2
```

Which ONE statement below best describes the result of executing this javascript code?

A) Car c will have an 'engine' property that has value `undefined`.
B) Variable `f` will be `undefined` because `c.set` does not invoke a function.
C) Line 2 will cause the global object to have an 'engine' property set to 'v8'.
D) Line 2 will cause Car c to have an 'engine' property set to 'v8'.
E) The code will result in a run-time error because function Car() should have parameters k,v.

10) Consider the following Javascript code:

```
function Car(){
   let that = this
   this.set = function(k,v){that[k]=v}
}

let c = new Car()
let f = c.set
f('engine','v8')   //line 1
```

Which ONE statement below best describes the result of executing //line 1 will be?

A) Car c will have an 'engine' property created and set to 'v8'.
B) Variable f will refer to a new object with its 'engine' property set to 'v8'.
C) The code will result in a run-time error because no object was specified in the execution of f().
D) The global object will have an 'engine' property created and set to 'v8'.
E) The code will result in a run-time error because f will not be recognized as a function.

11) Suppose `x` is a javascript object and `x.prototype` returns an object `{"colour":"red"}`. Which ONE of the following is correct about `x`?

A) x is an object but not a function.
B) x is a function but cannot be used as a constructor as in y = new x()
C) x is a function and objects created like y = new x() will have an inherited property colour.
D) The expression x.colour will return 'red'
E) x is a function that MUST be passed a colour argument when invoked.

12) Consider the following javascript code:

```
(function(){
     var x = 100
     var foo = function() {x *= 2}
     foo()
     console.log(x)

})()
```

Which ONE of the following is TRUE about the code above**?**

A) The code is not legal because the function has no name.
B) The code is not legal because a function cannot invoke itself.
C) The code prevents x and foo from being visible outside the code construct.
D) The code is a javascript example of recursion and will result in foo calling itself.
E) The code will not be legal because var x is not visible within the function referred to by foo.

13) Which ONE of the following best explains what the `f.prototype` property of a function `f` refers to in javascript?

A) It is the inheritance prototype of object `f`.
B) It is the inheritance prototype of an object created with `new f()`.
C) It is the same object as the one obtained from the expression `f.__proto__`.
D) It is not defined when `f` is used as a constructor with `new`.
E) It is `undefined` because `.prototype` only applies to non-function objects.

14) Which of the following best describes the implication of closures in javascript?

A) Closures ensure that opening brackets are matched by closing brackets.
B) Closures ensure that local variables of functions are treated as `private`.
C) Closures refer to the fact that variable declarations are hoisted to the top of function scope.
D) Closures mean that variables defined outside a function are not visible to that function.
E) Closures enable local variables of a function to be accessed after the function execution has returned.

15) Consider the following javascript meant to implement a socket.io based chat server that allows clients on different browsers to chat with each other:

```
const http = require('http')
const ecStatic = require('ecstatic') //npm module
const PORT = 3000
const ROOT_DIR = 'html'
const server = http.createServer(ecStatic({root: __dirname + '/' + ROOT_DIR}))
const io = require('socket.io')(server) //npm socket.io capability

server.listen(PORT) //start server

io.on('connection', function(socket) {
  socket.emit('serverSays', 'You are connected to CHAT SERVER')
  socket.on('clientSays', function(data) {
    console.log('RECEIVED: ' + data)
    socket.emit('serverSays', data) //broadcast message to everyone
  })
})
```

Which ONE of the following best explains why this code will not work?

A) There is no static server code to serve the required client html and related static files.
B) The server would not be reachable on port 3000.
C) The received messages will not be broadcast to all connected clients.
D) The server only allows one client to connect at a time.
E) There is no handler function listening for individual client messages.

16) Consider an application that lets browser clients request arrays of words from files stored on a node.js server. The client types the name of the desired file in a text input field and then hits a **Submit** button. The following is the client-side javascript function that handles the button click.

```
function handleSubmitButton() {

  let userText = $('#userTextField').val(); //get user text from input field
  let textDiv = document.getElementById("text-area")
  textDiv.innerHTML = textDiv.innerHTML + `<p> ${userText}</p>`

  let userRequestObj = {text: userText} //object to send to server
  let userRequestJSON = JSON.stringify(userRequestObj)

  $.post("userText", userRequestJSON, function(data, status) {
    let responseObj = JSON.parse(data)
    words = responseObj.wordArray //replace word array with new words
  })
}
```

Which ONE of the following statements about the above code is FALSE?

A) The function uses the jQuery library functions instead of just native javascript.
B) The handler will append html paragraph elements to the client web page being shown.
C) The client and server exchange JSON formatted data with each other.
D) The words array is over-written even if the server does not sent word array data.
E) The client will only work if the server sets the response MIME type to be `application/json`.

17) Consider the following code that implements a Node.js server.

```
const http = require('http') //needed for http messaging
const fs = require('fs') //needed to read and write files
const url = require('url') //useful to parse URLs
const ROOT_DIR = 'html' //directory for serving files from

http.createServer(function (request,response){
     let urlObj = url.parse(request.url, true, false)
     console.log("PATHNAME: " + urlObj.pathname)
     console.log("REQUEST: " + ROOT_DIR + urlObj.pathname)

     fs.readFile(ROOT_DIR + urlObj.pathname, function(err,data){
       if(err){
          console.log('ERROR: ' + JSON.stringify(err))
          response.writeHead(404)
          response.end(JSON.stringify(err))
          return
        }
        response.writeHead(200, {'Content-Type': 'text/html'})
        response.end(data)
       })
 }).listen(3000)
```

Which ONE of the following statements about this code is TRUE?

A) The application serves static files from the 'www' directory on the server.
B) The file reading is synchronous.
C) The `function(request,response)` is invoked when `createServer(…)` is called in the code.
D) .html files will be served with the correct http MIME type.
E) The 'html' directory is where we would expect to find client-side static files to be served by this server.

18) Which ONE of the following statements about `.json` files is TRUE?

A) They can be loaded directly into javascript objects using a require() statement.
B) They contain the javascript used by an html page to supply things like the mouse event handlers.
C) They only apply to client-side javascript code, not server-side code.
D) They are a proprietary node.js file format and cannot be used outside of node.js applications.
E) They are the mechanism used to help ensure that the correct MIME types are served to HTTP clients.

19) Consider the following code intended to implement a Node.js based static server that could serve files similar to those found on our course web site.

```
const http = require('http')
const fs = require('fs')
const url = require('url')
const ROOT_DIR = 'public' //directory for static files

http.createServer(function (request,response){

     let urlObj = url.parse(request.url, true, false)
     fs.readFile(ROOT_DIR + urlObj.pathname, function(err,data){
       if(err){
          console.log('ERROR: ' + JSON.stringify(err))
          response.writeHead(404)
          response.end(JSON.stringify(err))
          return
        }
        response.writeHead(200, {'Content-Type': 'text/html'})
        response.end(data)
      })
 }).listen(3000)
```

Which ONE of the following best explains why it would NOT work as intended?

A) JSON.stringify() is not possible unless the code requires it first as in: `const JSON = require('json')`.
B) The `readFile()` function being called is NOT asynchronous.
C) The MIME content type for static .css files will not be set correctly.
D) The variables http, fs, and url should NOT be declared `const`.
E) The response code of 200 should not be used with content of type 'text/html'.

20) Consider the following code meant to read a file as part of a Node.js application.

```
const fs = require('fs')
const inputFilePath = "songs/output.txt"
fs.readFile(inputFilePath, 'utf8', function(err, data){
    console.log('now running callback function')
    if(err) console.log('FILE RE-READ ERROR')
    let fileData = JSON.parse(data)
  })
console.log(fileData.songs[0])
```

Which ONE of the following explains why this code would not work as the programmer likely intended?

A) No ROOT directory is specified for the file path.
B) readFile's callback function has no name to refer to.
C) `JSON.parse(data)` will not work without a `require('json')` statement.
D) `console.log(fileData.songs[0])` will be called before the file contents have been read.
E) `function(err, data)` will be called before the file contents are read.