

Name: \_\_\_\_\_

Student ID #: \_\_\_\_\_

## COMP 2406: Fundamentals of Web Applications

### Winter 2016 Mid-Term Exam

February 11, 2016

This exam has 6 pages (including this one). There are 15 questions worth a total of 30 points. Answer all questions in an exam booklet. This exam is closed: no books, notes, or electronic aids. **Most of the questions refer to the code of `examdemo`. This code is listed on pages 5 and 6 of this exam. You may want to detach this page for ease of reference.** When asked to fix portions of the program, you may explain the strategy for your fix (how you would do it) rather than give precise code. You have 80 minutes. Good luck!

1. (3) What is the type of the values associated with all of the symbols referenced in line `examforms.js:44`? Specifically, what is the type of the values bound to `server`, `on`, and `serverUp`?

**A: `server`: object, `on`: function, `serverUp`: function**

2. (2) If I create a line 46 in `examforms.js` (i.e., add a line to the end of the file) that says `'console.log("Finished!");'`, when will `finished!` be printed relative to the other output of the program? Explain your reasoning.

**A: "Finished!" should generally be printed first, before any other output, because all of the other output is produced by callbacks and they are normally executed after the main body of the file has been evaluated. It is acceptable to say that you don't actually know when it will be printed since you can't know for certain when callbacks will be invoked; however, the mainly single-threaded computation model of JavaScript makes this sort of interleaving very unlikely (unless you use something like web workers).**

3. (2) If you change the `app.post()` call to an `app.get()` call on `examforms.js:19`, how would the program no longer function properly? Is it possible to repair it *without* changing this line back to `app.get()`? (yes or no)

**A: It would no longer function properly because the form submission generates a "POST /add" HTTP request and that HTTP request is no longer recognized by the server. Yes. (If you change the form's method (`index.jade:8`) to `get` and make the function access parameters as passed by GETs (`req.query` rather than `req.body`) then it would be fixed.)**

4. (2) As written, examforms returns a 404 error for requests to style.css. What is wrong with examforms? How could you fix it?

**A: examforms has no way of serving static files. You'd either have to create a custom route that would load the style sheet manually, or you'd have to include the line normally used in express node applications to add static server support. (Specifically, you'd add `app.use(express.static(_dirname))`; around examforms.js:14.)**

5. (2) The request object (req) passed in to route handlers has a "headers" property, just as the request objects in tinywebserver.js. Given this fact, how could you change examforms so that it uses a different style sheet when receiving requests from iPhones? (Describe the general strategy, no need for all the implementation details.)

**A: We would just have to check the value of `req.headers["user-agent"]` to see if it contained the string "iPhone" in each route handling function. We'd then have to either use different jade files for iPhones that used the different stylesheet, or we'd move the stylesheet from layout.jade and instead pass it in like we do the title of the page.**

6. (2) When a web browser loads a standard web page, what type of HTTP request does it make of the server? And, if the requested URL is **invalid** (i.e., page not found), what is the standard response code?

**A: HTTP GET; 404**

7. (2) How could you change examforms so you could specify the PORT on the command line, e.g. you'd run `node examforms.js 7000` to have it listen on port 7000? If no port is specified it should listen on port 3000.

**A: Add a line to set the port variable based on the value of `process.argv[2]`, e.g. `var port = process.argv[2] — 3000`;**

8. (1) What is the purpose of the `Content-Type`: HTTP response header?

**A: It tells the browser the type of content being returned, whether it is a plain text file (text/plain), an HTML document (text/HTML), an image (image/jpeg), etc.**

9. (2) When the call to `app.get` on lines 15–17 of examforms.js returns, what work has it accomplished? Specifically what data (if any) has been returned to the requesting web client? Why?

**A: After these lines have been written a function has been defined and it has been registered as a callback for an HTTP GET / request. The anonymous function has not been called; it will only be invoked when there is an HTTP GET / request.**

10. (2) When is the function `serverDown()`, on lines 37-40 of examforms.js, called? How can a user cause this function to be called?

**A: This function is called when the server receives a SIGINT signal. This signal is generated when a user presses Ctrl-C at the command line.**

11. (2) What would happen to examforms if we deleted layout.jade:6 (block header)? What about layout.jade:8 (block content)?

**A: If you delete line 6, nothing would happen as this block isn't used in any of the templates. Deleting line 8 would make the body of all pages blank.**

12. (2) The `each` statement on list.jade:16 is a loop. What does this loop do? Where is it executed (server or browser)?

**A: This loop generates a table line for every person stored in the state array (passed to the template on examforms.js:30). It runs on the server as it is a Jade loop. The browser simply gets a fully populated table.**

13. (2) What would happen (if anything) if you removed two spaces from the beginning of list.jade:25, `button(type="submit") Home?`

**A: By removing these spaces the button is no longer indented under the form on line 24; thus now it is a submit button that is no longer enclosed inside of an HTML form. Thus while the button will still be present in the page, clicking it will now do nothing.**

14. (2) Do web browsers normally run/interpret Jade code? Explain briefly.

**A: Web browsers do not interpret Jade code; instead, they interpret the HTML that is produced by compiling Jade templates.**

15. (2) What does `res.redirect(/list);` on line 26 of examforms.js do? Specifically, what response is (eventually or immediately) sent back to the browser after this line is executed? Explain briefly.

**A: This line causes a 302 (permanent redirect) response to be sent back to the browser with /list as the new location. The browser then proceeds to do an HTTP GET /list which then causes the callback for /list to be run on the server (and return the corresponding page).**

---

This is extra space for answering questions.

### examforms.js:

```
1 var http = require('http');
2 var express = require('express');
3 var bodyParser = require('body-parser');
4 var logger = require('morgan');
5 var port = 3000;
6 var state = [];
7
8 var app = express();
9
10 app.set('view engine', 'jade');
11 app.set('views', __dirname);
12 app.use(logger('dev'));
13 app.use(bodyParser.urlencoded({ extended: false }));
14
15 app.get('/', function(req, res, next) {
16   res.render('index', { title: 'COMP 2406 Exam form demo' });
17 });
18
19 app.post('/add', function(req, res) {
20   var obj = { name: req.body.name,
21             city: req.body.city,
22               country: req.body.country,
23               birthday: req.body.birthday,
24               email: req.body.email };
25   state.push(obj);
26   res.redirect('/list');
27 });
28
29 app.get('/list', function(req, res) {
30   res.render('list', { title: 'People Listing', items: state});
31 });
32
33 var serverUp = function() {
34   console.log("ExamForms listening on port " + port);
35 }
36
37 var serverDown = function() {
38   console.log("Server shutting down.");
39   process.exit(0);
40 }
41
42 var server = http.createServer(app);
43 server.listen(port);
44 server.on('listening', serverUp);
45 process.on('SIGINT', serverDown);
```

### layout.jade:

```
1 doctype html
2 html
3   head
4     title= title
5     link(rel='stylesheet', href='/style.css')
6     block header
7   body
8     block content
```

### index.jade:

```
1 extends layout
2
3 block content
4   h1= title
5
6   div
7     p Fill out your info
8     form(method="post", action="/add")
9       div
10        input#name(type="text", name="name")
11        label Name
12      div
13        input#country(type="text", name="city")
14        label City
15      div
16        input#country(type="text", name="country")
17        label Country
18      div
19        input#birthday(type="text", name="birthday")
20        label Birthday
21      div
22        input#email(type="text", name="email")
23        label Email
24      button(type="submit") Submit
```

### list.jade:

```
1 extends layout
2
3 block content
4   h1= title
5
6   div
7     div
8       table
9         thead
10          th Name
11          th City
12          th Country
13          th Birthday
14          th Email
15        tbody
16          each item in items
17            tr
18              td #{item.name}
19              td #{item.city}
20              td #{item.country}
21              td #{item.birthday}
22              td #{item.email}
23
24      form(method="get", action="/")
25      button(type="submit") Home
```

### style.css:

```
1 body {
2   padding: 50px;
3   font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;
4 }
5 a {
6   color: #00B7FF;
7 }
```