IT 231 – Final Exam
Spring 2023

Name ________________________

Part A: Multiple Choice. Optionally, for each question, show your work or give a reason why you chose your answer. 5 points each.

1. Which of these Command Prompt (Windows) or Terminal Window (Mac) commands creates a Node.js project?

   a. `npm create` b. `npm init` c. `node create` d. `node init`

2. The server app is defined and configured like this in a basic Node project that does not use Express:

   ```
   const app = http.createServer( (req, res) => {
       if (q.url == "/greeting/first") {
           res.writeHead({"Content-Type"," text/html"});
           res.write("Hello, World.");
           res.end( );
       }
   });
   app.listen(3500);
   ```

   Which localhost URL is used to view the server content?

   a. `localhost:3000/`
   b. `localhost:3000/greeting/first`
   c. `localhost:3500/greeting/first`
   d. `localhost:4000/greeting/`

3. Which CSS property sets the distance between an HTML element and its border?

   a. `distance` b. `gap` c. `margin` d. `padding`

4. The function f is defined as

   ```
   var f = (a, b) => 3 + 4 * a + 5 * b?
   ```

   What is the return value of `f(2, 0)`?

   a. `3` b. `11` c. `12` d. `13`

5. In an Express project, in which folder are the files with the `.ejs` extension located?

   a. `__dirname` b. `ejs` c. `ejs-files` d. `views`

6. The array `arr` is defined as

   ```
   var arr = ["apple", "orange", "peach", "grapefruit"];
   ```

   Which statement removes `"orange"` and `"peach"` from the array and inserts `"lime"` and `"lemon"`?

a. `arr.splice(1, 2, "lime", "lemon");`

b. `arr.splice(1, 2, ["lime", "lemon"]);`

c. `arr.splice(1, ["orange", "peach"], ["lime", "lemon"]);`

d. `arr.splice(2, 2, "lime", "lemon");`

7. Consider the CSS class:

```
.j1 { font-family: Jokerman;  font-weight: bold;
      font-size: 133%; }
```

How do you apply the this class to the h1 header

`<h1>Sample Header</h1>`

The result should be

Sample Header

a. `<h1 class=".j1">Sample Header</h1>`

b. `<h1 class="j1">Sample Header</h1>`

c. `<h1 style=".j1">Sample Header</h1>`

d. `<h1 style="j1">Sample Header</h1>`

8. An express route is set up to accept a parameter like this:

`app.get("specials/:item", callbackFunction);`

How can the parameter value be obtained to use in the `main.js` script?

a. `var item = req.param:item;`   b. `var item = req.params.item;`

c. `var item = res.param.item;`   d. `var item = res.params.item;`

9. Which Embedded JavaScript statements create a table with one column that displays the numbers from 1 to 100?

a.
```
<table>
  <% for(let i = 1; i <= 100; i++) { %>
     <table>
         <tr> <td> <%= i %> </td> </tr>
     </table>
  <% } %>
```

b.
```
<table>
     <tr> <% for(let i = 1; i <= 100; i++) { %>
         <td> <%= i %> </td>
     <% } %> </tr>
 </table>
```

c.
```
<table>
     <%= for(let i = 1; i <= 100; i++) { =%>
         <tr> <td> <% i %> </td> </tr>
     }
```

```
 </table>
```

d.
```
<table>
     <% for(let i = 1; i <= 100; i++) { %>
         <tr> <td> <%= i %> </td> </tr>
     <% } %>
 </table>
```

10. The following form, defined in the **`infoform.ejs`** file is used to enter the full name and phone number for a customer:

```
<form name="form1" action="/receipt" method="get">
      <label for="fullName">Full Name:</label><br>
      <input type="text" name="fullname"><br><br>
      <label for="phone">Phone:</label><br>
      <input type="text" name="phone"><br><br>
      <input type="submit" value="Submit Contact Information">
</form>
```

Also, this **`app.get`** method in **`main.js`** sets up the **`/receipt`** route:

```
app.get("/receipt", (req, res) => {
      var contactInfo = req.query;
      res.render("receipt", {contactInfo: req.query});
});
```

What is the correct EJS code in the **`receipt.ejs`** file to display the contact information?

a.
```
<h1>Contact Information</h1>
<ul>
    <li>Full Name: <%= contactInfo.fullname %></li>
    <li>Phone Number: <%= contactInfo.phone %></li>
</ul>
```

b.
```
<ul>
    <li>Full Name: <% contactInfo.fullname %></li>
    <li>Phone Number: <% contactInfo.phone %></li>
</ul>
```

c.
```
<ul>
    <li>Full Name: <% contactInfo.fullname %></li>
    <li><% Phone Number: contactInfo.phone </li> %>
</ul>
```

d.
```
<ul>
    <li>Full Name: <%= infoForm.fullname %></li>
    <li>Phone Number: <%= infoForm.phone %></li>
    </ul>
```

Part B. Short Essay. Answer 2 out of 3 questions. For full credit, write using paragraphs with an introduction and a conclusion. One third to one half of a page per short essay. 10 points each.

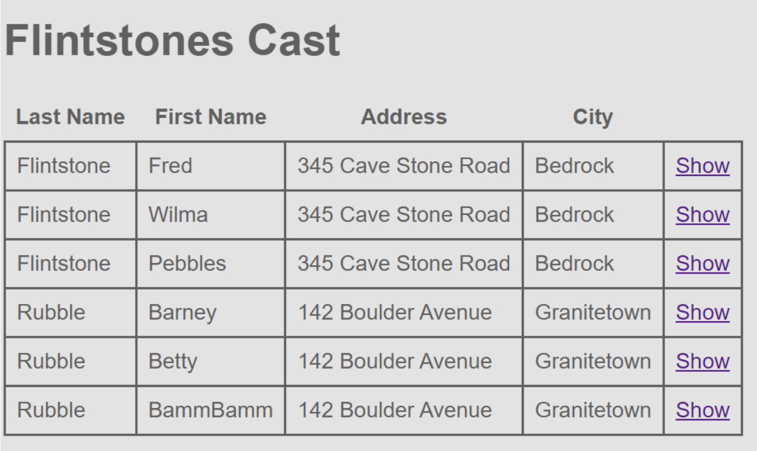
1. Explain how the projects in IT 130 differ from the projects in IT 231. Target your explanation to someone that knows HTML and CSS, but not Node and Express.

2. Explain how Node differs from Express. Give examples.

3. What is HTTP? Explain the differences between the GET and the POST commands.

Part C: Find the errors. The flintstones-cast project is an Express project that reads data from the data/flintstones-cast.json file and displays it on index and show views.

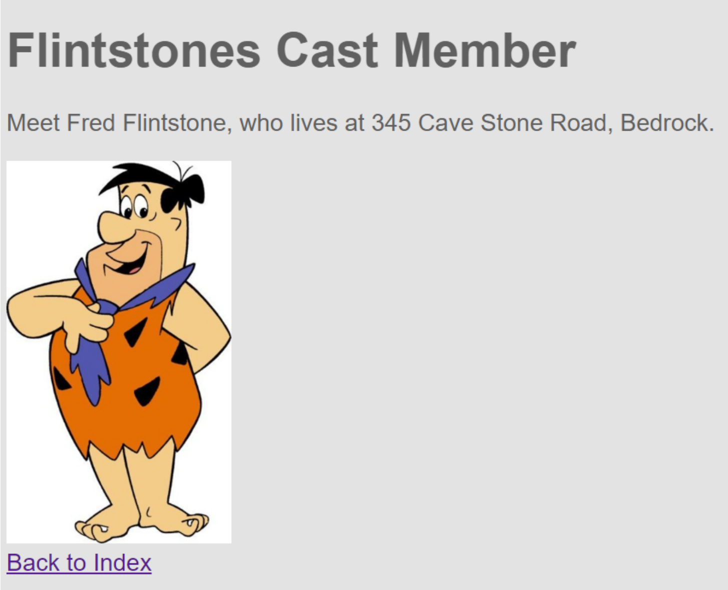
The index view that is displayed using the route `/flintstones_cast`



Flintstones Cast

Last Name	First Name	Address	City	
Flintstone	Fred	345 Cave Stone Road	Bedrock	Show
Flintstone	Wilma	345 Cave Stone Road	Bedrock	Show
Flintstone	Pebbles	345 Cave Stone Road	Bedrock	Show
Rubble	Barney	142 Boulder Avenue	Granitetown	Show
Rubble	Betty	142 Boulder Avenue	Granitetown	Show
Rubble	BammBamm	142 Boulder Avenue	Granitetown	Show

The show view that is displayed using the route `/flintstones_cast/show/0`
The parameters 0 through 5 refer to rows in the preceding index table.



Flintstones Cast Member

Meet Fred Flintstone, who lives at 345 Cave Stone Road, Bedrock.

Back to Index

Pages 6 through 9 show the source code files for the flintstones-cast project. However, there are errors in each source code file. Correct the errors directly on the source code pages; do not recopy. Adding or correcting the following pairs of symbols only counts as one error: braces (`{ }`), brackets (`[ ]`), parentheses (`( )`), angle brackets (< >), single quotes (`' '`), double quotes (`" "`), backquotes (`` ` ` ``). 15 points.

```
//-------------------------------------------------------------
// Source code file main.js

const fs = require("fs");
const express = require(express);
const app = express;
const PORT = 3500;

// Load data from file
fs.readFile("data/flintstones-cast.json", (err, data) => {
    if(err) {
        console.log("Error reading flintstones-cast.json.");
        process.exit(1);
    }
    var cast = JSON.parse(data);
});

app.use("view engine", "ejs");
app.set(express.static("public"));

app.get("/flintstones_cast", (req, res) => {
     res.render("index", { cast: cast });
});

app.get("/flintstones_cast/show/i", (req, res) => {
    var i = res.params.i;
    if (0 <= i && i < cast.length) {
        res.render("show", { cast: cast, i: i });
    }
    else {
        res.render("<p>Array index out of bounds.</p>");
    }
});

app.listen(PORT, ( ) => {
    console.log("Server listening on PORT ${PORT}.");
});

//-------------------------------------------------------------
```

```
--------------------------------------------------------------

// Data file: data/flintstones-case.json

[ {"lname": "Flintstone", "fname": "Fred",
   "addr": "345 Cave Stone Road", "city": "Bedrock"},
  {"lname": "Flintstone", "fname": "Wilma",
   "addr": "345 Cave Stone Road", "city": "Bedrock"},
  {"lname": "Flintstone", "fname"; "Pebbles",
   "addr": "345 Cave Stone Road", "city": "Bedrock"},
  {"lname": "Rubble", "fname": "Barney"
   "addr": "142 Boulder Avenue", "city": "Granitetown",
  {"lname": "Rubble", "fname": "Betty",
   "addr": '142 Boulder Avenue', "city": "Granitetown"},
  {"lname": "Rubble", "fname": "BammBamm",
   "addr": "142 Boulder Avenue", "city": "Granitetown"}

--------------------------------------------------------------


/* Source code file: public/css/styles.css */

body { backgroundColor: 606060;
       color: #606060;
       font: 100%, helvetica sans-serif; }
table { border-collapse: collapse; }
td { border: 2px #606060 solid; }
td th { padding: 8px; }
img { width: 150in; }

--------------------------------------------------------------
```

```
--------------------------------------------------------------

<!DOCTYPE html>

<!-- Source code file: views/index.ejs

<html lang="english">
    <head>
        <meta charset="UTF-8">
        <title>Flintstones Cast
        <link rel="stylesheet" href="/css/styles.css">
    </head>

    <body>
        <h1>Flintstones Cast</h1>
        <table>
            <tr>
                <th>Last Name</th>
                <th>First Name</th>
                <th>Address</th>
                <th>City</th>
            </tr>
            <% var index = 0;
               for(person : cast) {
                   <tr>
                       <td><%= person.lname %></td>
                       <td><%= person.fname %></td>
                       <td><% person.addr %></td>
                       <td><%= person.city %></td>
   <td><a href="/flintstones_cast/show/<%= index %>">Show</td>

                  </tr>
              <% index--;
                 } %>
        </table>
    </body>
</html>

--------------------------------------------------------------
```

```
--------------------------------------------------------------

<!DOCTYPE html>

<!-- Source code file: views/show.ejs -->
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Flintstones Cast</title>
        <link rel="stylesheet" href="/css/styles.css">
    </head>

    <body>
        <% var person = cast[i]; %>

        <h1>Flintstones Cast Member</h1>
        <p>Meet <%= person.fname %> <%= person['lname'] %>,
           who lives at <%= person.addr %>,
                        <%= person.city %>.</p>
        <img src="/images<%= person.fname %>.jpg"
             alt="Image of <%= person.fname %>"><br>
        <a src="/flintstones_cast">Back to Index</a>
    </html>
</body>


--------------------------------------------------------------
```