**Midterm Review**

**Format**

**The exam will be 1 hour and 25 minutes and will be closed note**.

\*\*If you have an ADS form and intend on using your accommodation, you must schedule your exam with the [testing center](https://www.towson.edu/accessibility-disability-services/testing.html). Due to scheduling conflicts, you will not be able to get accommodations if the exam is taken in the classroom at the regular time. Please schedule this a week in advance to avoid any issues.

**Material**

1. JavaScript/Node.js Overview:
   1. The difference between JavaScript and Node.js
   2. How to run a JavaScript file in Node.js
   3. What is npm and its purpose
2. Variables
   1. Describe how variables are “typed” in JavaScript
      1. The operator that determines the type
   2. The different data types in JavaScript and know how many bytes are in each
      1. Number
      2. Boolean
      3. Null
      4. String
         1. Be able to use String functions/properties in a coding question
            1. length
            2. toUpperCase() and toLowerCase()
            3. charAt()
            4. replace()
            5. split()
         2. Be able to loop through the characters of a String
3. Data Types
   1. Arrays
      1. Building and manipulating arrays using functions/properties
         1. push()
         2. pop()
         3. concat()
         4. length
      2. Looping though arrays
   2. JSON
      1. Creating your own JSON object
      2. Given a large JSON object, be able to access different keys
4. Process.argv (see lecture 2)
   1. Be able to create a program that takes in command line arguments
   2. Be able to match commands with the correct indexes
5. Conditionals
   1. Be able to write simple if statements
   2. Know the difference between == and ===
6. Functions
   1. Be able to write simple functions, potentially with parameters and/or return statements
   2. Anonymous functions
      1. Be able to write an anonymous function set equal to a variable
      2. Be able to pass an anonymous function to another function
      3. Be able to define an anonymous function in-line (lecture 3 slide 16)
   3. forEach
      1. Be able to use a forEach to iterate through an array and perform some action on it
      2. Be able to use the three parameters that a forEach provides
7. Asynchronous JavaScript (Lecture 4 video)
   1. How Synchronous functions are executed (callstack)
   2. How Asynchronous functions are executed (webapis-taskqueue-callstack)
      1. Why callbacks are necessary in asynchronous programming
   3. setTimeout function
8. Node Modules
   1. How to import a Node Module
   2. Be able to write a custom Node Module with asynchronous properties (similar to the lecture 5 example)
9. Express.js
   1. Be able to write a simple Express app
   2. app.get()
      1. The parameters of app.get():
         1. The path
            1. Be able to use different paths other than root (/)
         2. A callback function(req, res)
            1. req is a JSON object from the client
            2. res will be the response we (the server) send to the client

res.send() – sending different types of data

* 1. app.listen()

1. RESTful API:
   1. At a high-level explain what an API is
      1. Be able to give examples of an API
   2. Be able to explain what an “endpoint”
   3. Be able to create a server using the HTTP module
      1. res.end()
      2. server.listen()
   4. url modue
      1. Obtaining the url from a requester
      2. Obtaining the parameters from a url – parse
         1. What happens when url.parse is true
         2. What happens when url.parse is false
   5. Status codes
      1. 200 status code meaning
      2. 400 status code meaning
      3. res.writeHead() function
         1. JSON.stringify() purpose
2. HTTP Request:
   1. Difference between a GET and POST
   2. Explain the concept of a HTTP request (how it is performed at a high-level)
   3. Be able to use the request function (from the request module)
      1. Be able to explain:
         1. Err
         2. Response
         3. Body
   4. Be able to perform a GET request on a given url by passing data through the url
   5. Be able to perform a GET request on a given url by passing data through a Query String
   6. JSON.parse()
   7. Explain oAuth and its importance
   8. Be able to perform a simple POST
3. Git & AWS:
   1. Explain Git, its importance, and how it is different than the predecessor version control systems
   2. Be able to know the following git commands
      1. add
      2. commit –m “message”
      3. push origin <branch-name>
      4. clone
      5. status
      6. branch –a –v
      7. git checkout –b <branch-name>
   3. Explain why branching is important and the advantages
   4. Explain what an EC2 instance is
   5. Explain what a S3 instance is
   6. Describe when a web developer would use each instance
4. MongoDB:
   1. Explain the advantages of using MongoDB
   2. Be able to create a Mongo Collection
   3. Be able to insert a Document into a given Collection
      1. Be able to insertMany as well
   4. Be able to find all Documents with a certain key(s)
   5. Be able to sort
      1. Ascending
      2. Descending
   6. Be able to drop a Collection
   7. Be able to update a Document within a Collection