**Study Guide for CS 363 Test**

**Exercise 1**

Revisit your solution to lab exercise 1.1, addressing the following questions along the way.

Compare and contrast:

The Internet

The World-Wide Web

**The Internet is comprised of nodes, that connect to each other. The world-wide web is the addresses after http:/... The way the World-Wide web is constructed is it often uses the Internet as a protocol.**

Compare and contrast SPAs with:

Traditional webpages

Desktop applications

**SPA’s (Single Page Applications) are like Desktop Applications in that they both strive to put the heavy lifting and processing on the client’s side. The button presses, the interactions are dealt with on the clients side. SPA’s are like webpages, in that they both are Internet protocols based in HTTP/HTPPS. SPA’s are unlike Desktop Applications, because one is obviously online, and thus is “normalized” across web browsers. SPA’s are unlike traditional webpages because SPA’s focus on putting the control in the clients hands, where traditional webpages, put the control of the web app in the servers hands. Performance wise, this particular comparisons differences can shine through.**

**Exercise 2**

Is JavaScript a “kluge”? a “sloppy” language? If so, give examples of why; if not, explain why not.

**JavaScript is considered a “sloppy” language because the nature of the language is not meant to build objects, or traditional reasons. JavaScript was meant to be used/integrated into websites, and as such it’s syntax structure seems a bit sloppy. One example is the use of functions. If you want to define a function, there is no set way to do it. You can do it two ways: Anonymous Functions. In normal language every function needs an identifier (name), but in javascript you don't.**

Review the JavaScript features listed in Guide 2.1.a.

**First-class Functions- Functions that have variable names.**

**Anonymous Functions- Functions that have no name.**

**Polymorphism- Inheriting objects. Ex. A person object and a Student Object....**

**Closures- Calling an inner function in a outer function.**

**‘Strict Mode’- A tighter syntax checker mode. Normal silent errors, would throw errors and stop programs. It speeds up execution.**

**This keyword- The ‘this’ keyword lets you access private member variables in the program.**

**Promises- A object that describes a eventful completion or failure of asynchronous operation.**

**Exercise 3**

What are some web development stacks?

**Web Stack, are software that are required for Web Development. One web devolpment stacks are LAMP. LAMP has Linux, Apache for web server, MySQL for relational databases, PHP is the object oriented scripting language.**

What is the value of choosing Node.js?

1. **It works on all major platforms, Microshit, Macinshit, and Linux.**
2. **It is event-driven.**
3. **Non-blocking I/O that makes it lightweight and efficent**
4. **No Buffering, data is received in chunks.**

Consider the sample Node.js server code. What evidence do you see of the key features of JavaScript and of Node.js?

**-First-Class functions, the http.createserver, is a function given to a variable const.**

**-There are anonymous functions () => {} with the server.listen....**

Consider the sample server’s use of Node.js modules. Of what value are modules and NPM?

**The modules have the dependencies for the server to run. The NPM is the package manager that is used to download dependencies for the server to run.**

1. As we May Think
   1. To what war is Bush referring?

**World War Two**

* 1. With what problem is Bush concerned and how does he propose to solve it?

**The idea of extending human memory and human ability, to catalogue categorize and recollect the totality of human information.**

**He proposes the memex, a device with two screens,levers, and a magnetic tape (Computer server for a website), data would be linked to each other.**

* 1. Is there a modern-day equivalent to the memex?

**The World-Wide web. That is to say the magnetic strips are hard drives, and the links, are trails of associated information, one could create and share collections of connected data which would not only be more intuitive.**

1. Web Application Architecture from 10,000 Feet
   1. Why would one want to build a Single-Page Application (SPA)?

**Most of the data (code JS,HTML,CSS) is handled by the client, which means there isn’t latency and page hangups. Which means clients would be happier.**

* 1. What makes SPAs different from traditional applications?

**SPA’s have it so that the client has all the code, while the database is on the server side. So it makes the client feel like a native app, much like apps on your phone.**

* 1. What is the “dirty secret” of SPA development?

**HTML,CSS, and JavaScript were not designed for creating GUIs.**

**Frameworks- Use them as ways to build on your code, without having to deal with many problems.**

**Libraries- Use as a building block to get ceratin parts of your application to work as desired.**

1. Comparison of web browser engines
   1. What does a browser engine do?

**Browser Engine- turns the object that contains the HTML CSS and JS code, into a visually displayed resouce on the users device.**

* 1. What’s the most popular browser?

**CHROME**

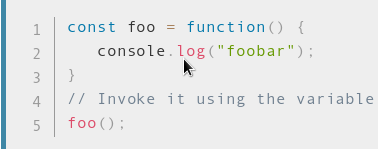
* 1. What engine does Google Chrome use and how was it built? (See the Wikipedia article for details.)

**Webkit Broswer engine.**

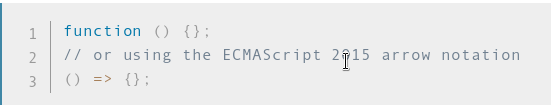
**It was built by u seing over 25 different code libraries from Google and Firefox. It was orginally made for small programs, but supports its own applications better.**

[JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript" \t "/home/justin/Documents\\x/_blank)

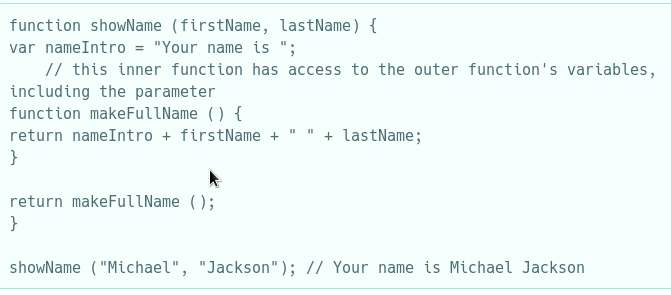
* 1. Explain the following concepts:
     1. first-class functions
        1. **Functions are treated like variables. , they have names of functions and have names for variables**



* + 1. anonymous functions
       1. **Functions without a name**



* + 1. prototype-based inheritance/polymorphism
       1. **Making objects, and sub-objects that inherit previous objects.**
          1. **Ex. Create a Animal object, then create a bird object, or bee object...**
    2. Closures
       1. **Letting a inner function access outer functions variables.**



* + 1. strict mode
       1. **Can be used to speed up code.**
       2. **Making silent errors throw errors.**
    2. this keyword (see Guide→Working with Objects→Using this for object references)
       1. **You can use it access a private variable within the class definition.**
    3. promises (see Guide→Using Promises)
       1. **The eventual completion of an asynchronous operation.**
  1. Compare and contrast the “sameness” operators: ==, ===
     1. **Compare: They both check for ‘equality’.**
     2. **Contrast: == is not strict, === is strict.**
  2. Compare and contrast JavaScript and Java.
     1. **Compare: Both are OOP based.**
     2. **Contrast: Java is used for apps and can be used independent of a web browser. JavaScript cannot be separated from he browser, and is integrated in the browser.**
  3. Compare and contrast ES5 and ES6 (aka. ES2015).
     1. **Compare: They both are based in JavaScript.**
     2. **Contrast: ES5 has a lot of high-level language shortcomings.**
     3. **Contrast: ES6 is a lot more like other high-level languages.**

[The World's Most Misunderstood Programming Language](http://crockford.com/javascript/javascript.html" \t "/home/justin/Documents\\x/_blank)

* 1. Why is JavaScript considered such a “kluge”?
     1. **One reason is that the “script” part of the language indicates that the language less of a programming language and more like typescript language. This “-script” language indicates it has parts that are hopefully used together to make a purpose.**
     2. **Most of the people who program in it are not really programmers, they are web-developers who integrate its use in webpages.**
  2. Is this reputation justified?
     1. **I think yes, because the earlier versions of JS were buggy, with variable type errors like “ “1” + “1”.**

1. Node.js
   1. What features of Node.js make is attractive for building web servers?
      1. **It works on all major platforms, Microshit, Macinshit, and Linux.**
      2. **It is event-driven.**
      3. **Non-blocking I/O that makes it lightweight and efficent**
      4. **No Buffering, data is received in chunks.**
   2. When is it advisable to use Node.js, for I/O-bound operations or CPU-intensive operations? Why?
      1. **I/O bound operations, because of it’s event driven, nature. It uses a single thread, with event looping.**
2. NPM
   1. What is the difference between NPM global and local installations.
      1. **Global installation, means NPM is available to all programs, and NPM local, means it’s available to just a selected project.**
   2. What are Node modules and how does one use them?
      1. **Modules are sub-packages that you can use that provide difficult functionality, such as one letting you use parse URL’s, one lets you connect to a server.**
3. Express
   1. What are the layers of a Web application and what does each do?
      1. **Presentation Tier- The front end, it has the HTML, CSS, and JavaScript.**
      2. **Application Tier - business logic, and applications core abilities.**
      3. **Data Tier - database/data storage tier**
   2. Define and explain the following terms: middleware; routing; static files.
      1. **Middleware- The “middle”, in android apps, the android OS is the middleware, that is it separates the Linux kernel from the android apps.**
      2. **Routing- How a network packet, goes from its server to the client.**
      3. **Static- A sort of predefined method for packets to take from Server to client.**