Week 1 – Study Material – WWW component of WMDD 4820

Memorize the following:

1. Problem: a problem in general terms is a matter or situation that needs to be dealt with or overcome (solved). In IT (or programming), it general takes the for of inquiries, given conditions, requirements, reports, user interfaces, etc.

2. Problem Solving: is the process of finding solutions to sometimes difficult and complex issues (problems). Essentially, it's about gathering enough information to fully understand the situation or issue, working through the details and reaching a solution. It tends to involve a high level of critical thinking.

3. Steps in Effective Problem Solving:
    - Identify and understand the Problem
    - Generate a Clear Definition of the Problem and get it Verified
    - Define realistic/attainable Goals and/or Desired Outcome
    - Brainstorm Solutions (never just one solution – at least 3)
    - Analyze and Assess the Solutions and establish Pros v Cons for them
    - Choose a Solution
    - Rapid Protype the Solution
    - Evaluate to Ensure it actually Solves the Problem – if not, pick another solution and loop
    - Implement the Development and Application of the Solution

4. Useful Problem-Solving Skills:
    - Active Listening
    - Understanding the Business, Environment, Issues, etc.
    - Communication
    - Effective Analysis
    - Open Mindedness
    - Willing to be Wrong
    - Effective Research
    - Creativity
    - Decisiveness (Decision Making Skills)

5. Habits or Personality Traits to Overcome in order to be a better Problem Solver:
    - Hero Complex
    - Jumping Straight to developing a solution without Understanding the Problem
    - Ego/Overactive Pride/Stubbornness
    - Narcissism
    - Needing to be right
    - Fear of Failing or being wrong
    - Assuming instead of Understanding
    - Resistance to putting the time in to do effective Research
    - Closed Mindedness
    - Lack of Willingness to consider other people's ideas or solutions
    - Not allowing others to Express themselves or Communicate their Ideas

Week 2 Preparation – Key Terminology:

- Variables
  - Variables in a programming context are containers that hold values that will be used elsewhere in the program
  - As the name implies, the contents of these variables can change over time
  - Variable are declared via the JavaScript “let” command
- Constants
  - Constants in a programming context are containers that hold values that will be used elsewhere in the program
  - As the name implies, the contents of these constants cannot be changed
  - Constants are declared via the JavaScript “const” command
- Scope
  - Scope refers to the accessibility of a variable – in essence, it is the region of the program that has access to the variable or constant
  - Both the “let” and “const” commands create containers with “block” level scoping
  - With scoping, inheritance flows down to children, but not up to parents – for example:

```
let parVar = 25;

let testFunc = () => {
  let newVar = parVar * 2; // parVar is valid here
}

console.log(newVar); // This is invalid because newVar is out of scope
```

- JavaScript Variable Types
  - JS Variables are loosely typed – meaning the type of information stored in a variable can change over time – this is very different than most languages
  - JS Variable types are:
    - Number (64 bit floating-point)
    - String
    - Boolean (True or False)
    - Null
    - Undefined
    - Bigint
    - Symbol
  - Another type of JS Variable is the Object (Data Structures, Functions, Arrays, etc.)
- Assignment Operator
  - The equal sign (“=”) serves multiple purposes in JavaScript, but the equal sign by itself (“=”) only serves one purpose – to assign whatever’s on the right of it into whatever’s on the left of it.
  - The equal sign (“=”) by itself will always function as an assignment operator and it will never function as a comparison operator – for example:

```
if (x = y) ...
```

this statement will place the content of y, into the x container, and then always produce a result of “true” because the container x exists…this is not the intended outcome of this process, but because the programmer used an assignment operator instead of a comparison operator (“==” or “===”), this is how js will interpret the statement

- Variable Precedence:

Precedence	Operator type	Associativity	Individual operators
20	Grouping	n/a	(…)
18	new (without argument list)	right-to-left	new …
17	Postfix Increment	n/a	… ++
	Postfix Decrement		… --
16	Logical NOT	right-to-left	! …
	Prefix Increment		++ …
	Prefix Decrement		-- …
	typeof		typeof …
15	Exponentiation	right-to-left	… ** …
14	Multiplication	left-to-right	… * …
	Division		… / …
	Remainder		… % …
13	Addition	left-to-right	… + …
	Subtraction		… - …
11	Less Than	left-to-right	… < …
	Less Than Or Equal		… <= …
	Greater Than		… > …
	Greater Than Or Equal		… >= …
10	Equality	left-to-right	… == …
	Inequality		… != …
	Strict Equality		… === …
	Strict Inequality		… !== …
6	Logical AND	left-to-right	… && …
5	Logical OR	left-to-right	… || …
3	Assignment	right-to-left	… = …
			… += …
			… -= …
			… **= …
			… *= …
			… /= …
			… %= …
			… <<= …

			… >>= …
			… >>>= …
			… &= …
			… ^= …
			… |= …
1	Comma / Sequence	left-to-right	… , …