**Purpose**

The purpose of unit 05 was to test our combined knowledge for this quarter.

**Concepts**

**Global** – Oftentimes in programming we require certain variables or functions to be available to other pieces of code. In this class so far, we have written most of our variables in a global scope. This can be helpful when needing to access those variables inside of a loop without having use getters and setters on objects. In my opinion, damn near everything should be global since it not only simplifies execution, it also makes programs easier to read and understand. You may argue that this is bad programming practice, and potentially unsecure, and you are right.

**Frameworks** – Programming frameworks are a collection of libraries that are most often created by well meaning programmers that desire an easier wrapper around complicated functions or tedious work. It is my understanding that often, frameworks are unnecessary, hurt future compatibility, and decrease ease of use. I do recognize the benefits of simple libraries and prefer to use modular portable code when I can.

**Pointer** – In a good programming language, programmers have access to pointers, or references to an address in memory. This is an incredible useful feature when you need to quickly read or write a block of memory and do not desire to make a copy first. JavaScript for example has no concept of pointers, and this can make things difficult. (JavaScript)

**Compiled Code** - If you need a program to be fast, then compile it. This simply reduces the programs contents as close to machine code (binary) as possible. It reduces the amount of overhead required to load the program into memory and execute it, which decreases CPU work time and increases performance. It used to be that there was a penalty to compiled code since it could take a long time to reduce it to machine language, but CPU’s are much faster than they used to be. Only huge projects still suffer from long compile time.

**Just in Time Compilation** – This technique allows programs to be compiled at runtime instead of precompiled. Languages like Java and JavaScript use JIT. The benefit of a faster compile time makes the program slower than a precompiled one. However, this can be extremely beneficial when programmers want the compiler to not be a part of their workflow and allows them to make and test changes quickly.

**Implications**

Modern languages irk me, don’t even get me started on frameworks like React. I believe that programming should be taught with and editor and compiler and nothing more. Too often new programmers are enticed by the idea of building magical things using magical solutions. This just leads to problems later for instant gratification now, and leaves gaps in their education that is necessary for entering the workforce.

Often the code that new programmers will be working on will be legacy code, not shiny new toys. Without a solid foundation those toys lose their luster and fall apart.