**Chapter 11 Written Assignment:**

1. Define Closure. Show an example, in JavaScript, of a program with a closure from chapter 11 that is not the lab. Explain how the program works.
   1. Closures are a function along with a referencing environment
   2. A closure captures the value of variables in scope at the time the closure is created.
   3. The closure contains the actual environment, not a copy.
   4. Example:

Graphical user interface, diagram

Description automatically generated with medium confidence

1. Implement the following function that takes a password argument and returns a function that accepts a password guess and returns true if the guess matches the password, with a closure.

function makePassword ( **password** ) {

return **function guess(passwordGuess)** {

return (**passwordGuess === password** );

};

}

**var tryGuess = makePassword(“secret”);**

**console.log(“Guessing “nope”: “ + tryGuess(“nope”));**

**console.log(“Guessing ‘secret’: ” + tryGuess(“secret”));**

1. Implement a program in JavaScript that bakes a cake and sets a timer for 10 minutes.
   1. setTimeout( function () {

alert(“Time to take the cake out of the oven!”);

} , 600000);

1. In an 80s movie, computer animated characters were forced to fight in a computerized blood sport. The ones that died were referred to as derezzed. Name the movie.
   1. I actually don’t understand the question ☹
   2. Maybe I will need to read the chapter better but it is 11:57 PM already so I have to submit.
2. How do closures affect scope? Explain.

Text

Description automatically generated

1. What is an anonymous function? Write a line of JavaScript code that exemplifies an anonymous function.
   1. An anonymous function is a function expression that has no name.
   2. We basically will put the function, inline in the call to the setTimeout built in function.
   3. setTimeout(function() {alert ( “Time to take the cookies out of the oven”); }, 600000 );
2. What is a nested function? Show me an example of a nested function from Chapter 11.
   1. Nested functions are functions defined inside another function.
   2. A nested function local scope, just like other local variables.
   3. To bind the value of a variable in a nested function, use the value that’s defined in the closest enclosing function. If no value is found, then we should look in the global scope.
   4. Example:

Text

Description automatically generated

1. Can JavaScript determine the scope of a variable by reading the structure of code? If so, what is that called?
   1. Yes JavaScript can determine whether the variable is in local or global scope from just the structure of the code.
   2. This is called **Lexical scope.**
2. What is a free variable and why would you use it when coding?
   1. A functions typically has local variables in its code body and it also might **have variables that aren’t defined locally which is called free variables. The name free comes from the fact that within the function body, free variables aren’t bound to any values (not declared locally in the function).** When we have an environment that has a value for each of the  **free variables,**  we say that we’ve closed the function. And when we take the function and the environment together, we say we have a closure.
3. When is a function declaration defined? When is a function expression evaluated?
   1. A function declaration is defined before the rest of the code is evaluated.
   2. A function expression is evaluated at runtime with the rest of the code, and so it not defined until the statement in which it appears is evaluated.