1.

```
const compute = function(num) {
        total = total + num;
        return total;
}
```

Does this function have side effects? Yes, total is changed and we can see it is not a part of the scope.

2. Functions without side effects are considered pure. TRUE

3. A pure function depends on data outside its own scope. FALSE

4. What is one of the reasons to avoid side effects?
    A. It makes the code easier to test. CORRECT
    B. It makes for more compact code.
    C. It makes it easier to reason about. CORRECT
    D. It is more impressive to advanced programmers.

5. A JavaScript programs state is the current condition of the program, which is represented by data stored in variables and objects. TRUE

6. When state is shared, what is it shared between?
    A. Different programs.
    B. Two or more variables share the same data.
    C. Different scopes. CORRECT
    D. Two or more objects with the same data.

7. If an object contains the state of a program, and we want the state to flow through pure functions, what do we need to do to that object before passing it?
    A. Pull the data out into primitive values.
    B. Freeze the object so it can't be changed.
    C. Clone the object and pass the new object. CORRECT
    D. Share the object with the different functions.

8. What technique can be used to clone an object?
    A. Use JSON.stringify and then JSON .parse to first convert the object to a string and then convert it back to an object. CORRECT
    B. Assign the object to a different variable. Assigning an object copies the object.
    C. Use the Object.clone command.

9. TRUE or FALSE If you create an object obj and then assign that object to variable obj2. Then you change a property on obj2. The property on obj will not be changed. FALSE

10. If you plan to use the functional programming paradigm, don't mix it with other techniques you have learned. FALSE

11. The main difference between pipe and compose is the order in which you pass the functions you are composing. TRUE

12. What are the advantage(s) of function composition?
    A. Allows the Maintaining of simple, more reusable functions. CORRECT
    B. Allows the combining of simple functions to accomplish complex tasks. CORRECT
    C. Composing functions looks really cool.

D. It uses a more declarative style which is easier to read and reason about. CORRECT

13. Declarative code specifies how to do the computation instead of what to do. FALSE

14. In declarative code what happens to many of the control structures in the code?
    A. They appear more in order to account for the different approach.
    B. They are abstracted away with the functions we use. CORRECT
    C. They are still there but used differently.

1. If you find yourself calling the same function multiple times with similar parameters, you might want to use…
    1. Currying CORRECT
    2. Function composition
    3. Recursion
    4. Object cloning

2. Arity refers to what?
    A. Whether or not your functions are pure.
    B. The number of parameters in a function. CORRECT
    C. A function that acts as a procedure.
    D. A situation that requires currying.

3. A simple defintion of recursion is a function that calls itself. TRUE

4. Recursion works by placing each function call on the _____. Then when the final function is complete, it unwinds the _____.
    A. Load
    B. Watch Window
    C. Stack
    D. State

5. What part of a recursive function causes the recursion to end?
    A. Recursive Case
    B. End Case
    C. Base Case CORRECT

6. Ramda is a JS library that is designed for functional programming. TRUE