DWA_01.3 Knowledge Check_DWA1

---

1. Why is it important to manage complexity in Software?

The complexity of software development means it is easy to lose track of what is being coded, so it is important to make it readable for yourself and other developers to understand.. Because it is practically impossible to not end up with bugs in your code, one must write code such that they are able to identify errors immediately. This avoids catastrophic impact later on when the code has become complex.

Because of the complexity of softwares, their codebase is never rewritten but instead refactored, debugged and new code added to it. Therefore it is important to make it as readable and understandable as possible. Major softwares generally have long life cycles, meaning that they go through many changes made by a host developers.

---

2. What are the factors that create complexity in Software?

Build up of hidden bugs due to not adding code that prevents incorrect code from not working
Not naming variables in a descriptive and visibly unique manner.
Not making use of abstraction, leading to files with long code
Lack of documentation
Unreadable code style.

---

3. What are ways in which complexity can be managed in JavaScript?

Naming variable such that they are descriptive
Documenting:

- Adding extra information to variable names, e.g type of value
- Explaining functions
- Describing object shapes and behavior

Using code style that improve readability
Making use of abstraction
Add checks to prevent incorrect code from running
Adding tests to check for errors
Keep related entities close to one another (modular).

---

4. Are there implications of not managing complexity on a small scale?

Yes.
Code that is difficult to read makes it difficult to debug and improve, leading to time wasted on trying to understand the code and looking for bugs .

---

5. List a couple of codified style guide rules, and explain them in detail.

Name global constants in upper snake case. e.g CURRENT_POPULATION = 135 900

Be expressive in naming code.
For example, instead of using ‘time’ as a variable name, use ‘timeInSeconds’

Ensure variable names are visibly unique to avoid mistakenly using them incorrectly.

Use let or const to instead of var to declare variables. ‘let’ and ‘const’ are block scoped as opposed to function scoped as is ‘var’.
Use destructuring to access and make use of multiple properties of an object instead of creating multiple references. For example, use const { firstName, lastName } = user;
instead of const firstName = user.firstName;
  const lastName = user.lastName;

---

6. To date, what bug has taken you the longest to fix - why did it take so long?
A function that contained an abstracted function which had another abstracted function as its parameter was giving me errors. The reason being, i had not assigned the latter abstracted function to a variable and initialised another variable to which I assigned the latter abstracted function and write the variable to which I assigned the latter abstracted variable as a parameter to the variable to which I assigned the former abstracted function. The reason it took this long is that I did not see this as a problem as I had not really understood the importance of initialising variables and assigning functions to variables.
