

DWA_07.4 Knowledge Check_DWA7

1. Which were the three best abstractions, and why?

- Object-oriented programming: This is a way of thinking about programming that involves organizing code into classes and objects. This makes code easier to read, maintain, and reuse.
 - Functional programming: This is a style of programming that focuses on using functions to transform data. It's a very efficient way to write code and it can be easier to reason about than other styles.
 - Aspect-oriented programming: This is a newer way of thinking about programming that involves separating cross-cutting concerns into their own modules. This can make it easier to change parts of a program without affecting the whole thing. It's still being explored, but it's very promising.
-

2. Which were the three worst abstractions, and why?

- Callbacks: These are functions that are called in response to an event. They can be hard to reason about, especially when there are multiple callbacks being called in a specific order. They can also make code hard to debug and test.
 - Inheritance: This is a common feature of object-oriented programming. It can make code hard to read and understand, and it can lead to problems like the diamond problem. It can also make it hard to reuse code, since everything is tied to specific classes.
 - Monads: These are a type of data structure that can be used to represent other data structures. They can be very powerful, but they can also be hard to understand and use. They can also make code hard to read and maintain.
-

3. How can The three worst abstractions be improved via SOLID principles.

- Callbacks: One way to improve callbacks is by using a library like promises or async/await. These libraries make it easier to reason about callbacks, since they have a more intuitive syntax. They also make code easier to test.

- Inheritance: A way to improve inheritance is by using composition instead. With composition, you can reuse code without tying it to specific classes. This makes code easier to test and maintain.

monads: A way to improve monads is by using a library like Ramda. Ramda provides a set of functions that can be used to manipulate data without using monads directly. This makes code easier to read and understand.
