DWA_07.4 Knowledge Check_DWA7

---

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

1. homePreview() - The function required very little change due to the fact that it inherently was dynamically made to change depending on the data passed, making it easy to use it in various stages of the code.

2. colorThemePreference() - This function was very easy to understand and had a single use case, making it very easy to understand and use in a later stage.

3. filterOptions() - The majority of the functions code could have been replaced with previous functions, making it much simpler to understand.

---

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

1. setColorThemePreference() - Though most of the functions code was replaced with functions, some of the other code is dependent on outside data that cannot be provided by the functions listed.

2. bookPreview() - There was no change to this function, so there was not much effort in abstracting the content making this function debatably unusable in a modularity aspect.

3. showMore() - just like bookPreview(), the function had no change but it does have a single use case, but that single use case is required to be dynamically changed which was not applied in the function.

---

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

1. showMore() - using the Single-Responsibility Principle I dedicate the function to one singular purpose dynamically without an intake of unnecessary outside data.

2. bookPreview() - Interface Segregation Principle can help due to the fact that all the data being passed have all the same properties meaning all that must be done is that they have to abide by a particular blueprint.

3. setColorThemePreference() - The function does not really to be changed but merely extended upon to make it modular, which is why the Open-Closed Principle will solve a great deal of problems I may have.
