

DWA_07.4 Knowledge Check_DWA7

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

- A. **themeSettings** - This object follows the ***Single Responsibility Principle (SRP)*** because it has one responsibility: to store the color settings for each theme. It also follows the ***Open-Closed Principle (OCP)*** because new themes can easily be added, or existing themes can be modified without changing the object itself or the code that uses it.
 - B. **createBookPreview** function: This function follows the ***Single Responsibility Principle (SRP)*** because it has one responsibility: to create a book preview element. It also follows ***the Open-Closed Principle (OCP)*** because the structure or the or style of the book preview elements can be modified without changing the function itself or the code that calls it.
 - C. **updateBookList** function: This function follows the ***Single Responsibility Principle (SRP)*** because it has one responsibility: to update the book list. It also follows the ***Open-Closed Principle (OCP)*** because the way the book list is updated can be modified without changing the function itself or the code that calls it.
-

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

- A. **toggleOverlay function**: It's not very flexible because it only works with the settings overlay and assumes that the overlay is controlled by the open property.
 - B. **matches variable**: This global variable is used in multiple functions which can make the code harder to understand and maintain because you need to keep track of where and how this variable is used and modified.
 - C. **createOptionElement function**: also, not very flexible because it only works with option elements and assumes that the value and text are provided as parameters.
-

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

- A. **toggleOverlay function:** make this a generic function that takes the selector of the overlay and the state (open or closed) as parameters, so it could be used with any overlay and any method of showing or hiding overlays.
 - B. **matches variable:** Global variable used in multiple functions. A better approach might be to pass this variable as a parameter to the functions that need it, or to encapsulate it in an object or class that manages its state.
 - C. **createOptionElement function:** maybe make it a reusable function that takes the type of the element and an object of properties as parameters, so it could be used with any type of element and any properties.
-