What is Abstraction and Why is it important?

Abstraction is simplifying complex systems by hiding unnecessary details and exposing only important features. Abstraction allows developers to focus on what the object does rather than how it does it. For instance, one does not need to understand the detailed workings of the engine or braking system because you can interact with the wheel and other key parts.

A very significant benefit of abstraction in software development is that it improves code readability and maintenance by abstracting unnecessary details to ensure clean code. It also helps manage complexity by allowing developers to focus more on high-level design.

Abstraction can be used for developing GUIs and database management systems. In object-oriented programming, abstraction is used by providing blueprint for derived classes and defining methods.

As shown below, in my journal program I used abstraction to define a class structure called **PromptGenerator** that hides unnecessary implementation details.

A computer screen shot of text

AI-generated content may be incorrect.

The PromptGenerator can automatically generate prompts from an already created list. However, in fulfilling the principle of Abstraction, no user needs to know how the PromptGenerator works to access it’s behavior or attributes.