# Assignment: Principle of Abstraction

Abstraction in programming is the practice of simplifying complex systems by exposing only the essential features while hiding the underlying implementation details. It allows developers to interact with objects or methods without needing to understand every part of how they work internally.

One important benefit of abstraction is maintainability. When code is abstracted properly, it becomes easier to update or modify parts of the system without affecting other components. This results in cleaner and more modular code that is easier to test and reuse.

In my program, I applied abstraction through the Entry class. This class represents a journal entry with a prompt, a response, and the date it was created. The internal logic for formatting the entry is hidden inside the ToString method, which provides a clean output without exposing how the string is built.

public override string ToString()

{

return $"{Date.ToShortDateString()} - {Prompt}\n{Response}\n";

}

By calling ToString on an Entry object, other parts of the program can display the entry without knowing how the date is formatted or how the string is constructed. This makes the code more intuitive and prevents duplication.

Overall, abstraction is essential for building clear and scalable software. It allows developers to focus on what an object does instead of how it does it, similar to how we use a car without needing to understand the engine.