# Abstraction

Abstraction is a technique that allows us to focus on the essential features of an object, while ignoring its irrelevant details. It is also to reduce complicated systems to smaller, easier-to-manage parts in order to make them easy to build. In programming, abstraction is used to simplify complex ideas by breaking them down into smaller, more manageable modules.

One significant benefit of abstraction is that it enhances code re-usability and maintainability. Developers can easily design reusable components that can be utilized in different portions of a program by separating out the implementation specifics. This encourages simple and clean code and lessens code duplication.

An example of Abstraction is a graphic design program that allows us to draw different forms, such triangles, rectangles, and circles. In this case, abstraction enables us to represent these forms as separate classes, each of which captures its own special attributes and behaviors. Without having to understand all of the details of each shape's implementation, the application's user can interact with these shapes using high-level methods like "draw" or "resize".

A code example from the Journal program I wrote is the Prompt-generator Class. The code is

*public class Prompt-generator*

*{*

*private List<string> \_prompts;*

*public Prompt-generator()*

*{*

*\_prompts = new List<string>*

*{*

*"Who was the most interesting person I interacted with today?",*

*"What was the best part of my day?",*

*"How did I see the hand of the Lord in my life today?",*

*"What was the strongest emotion I felt today?",*

*"If I had one thing I could do over today, what would it be?"*

*};*

*}*

*public string GetRandomPrompt()*

*{*

*Random rand = new Random();*

*int index = rand.Next(\_prompts.Count);*

*return \_prompts[index];*

*}*

*}*

The PromptGenerator class, which contains every aspect of prompt management and presents a simplified interface to the outside world, is an example of abstraction. The variable \_prompts in this class contains the list of prompts and protects it from outside access. GetRandomPrompt() is the only public function that takes away the complexity of choosing a random prompt from the list, giving users an easy-to-use interface. The class promotes simplicity and makes it simple to reuse the prompts in other program sections.