# What is Abstraction?

Abstraction is about hiding the intricate details of how something works and focusing on the basic details only. This allows us to focus on what something does rather than how it does it. An important benefit of abstraction is that it simplifies the code, making it easier to understand, update, and reuse it. In object-oriented programming, we use abstraction with classes to manage different functions. For example, in the Journal Program, the Journal class is used to add and display entries. The Entry class manages individual entries for text and dates, while the Prompt Generator class creates prompts for new entries. This structure allows the program as a whole to interact with these classes in a simplified way, making the code easier to maintain and expand.

Below is an example of how entries are added and displayed by calling the relevant functions in the Journal class.

**Program.cs**

string option = Console.ReadLine();

            switch (option)

            {

                case "1":

                    WriteEntry();

                    break;

                case "2":

                    \_journal.DisplayEntries();

                    break;

**Journal.cs**

class Journal

{

    public List<Entry> \_entries = new List<Entry>();

    public void AddEntry(Entry newEntry)

    {

        \_entries.Add(newEntry);

    }

    public void DisplayEntries()

    {

        foreach (Entry entry in \_entries)

        {

            entry.Display();

        }

    }