Abstraction is a way of simplifying the world around us. In art, a person becomes a collection of simple geometric shapes. In code, it becomes the code we work with. Computers work with 1s and 0s. It can be long and difficult to read, much less execute. In order to make it easier to understand, we created languages for code that execute those lines using simpler language. Through this process we end up with our coding languages of today. Abstraction is helpful because it simplifies complex code into simple commands. In C#, Console.WriteLine(“Hello World!”); would normally require many more lines to write one line to the console. With that one line though, it carries out all of those functions.

Here is a better example of a project I’ve worked on recently.

public class Entry

{

    public string date { get; set; }

    public string prompt { get; set; }

    public string text { get; set; }

    public Entry(string currentDate, string givenPrompt, string userText)

    {

        date = currentDate;

        prompt = givenPrompt;

        text = userText;

    }

    public void Display()

    {

        Console.WriteLine($"Date: {date} - Prompt: {prompt}");

        Console.WriteLine(text);

    }

}

This class is another form of abstraction. Each of these lines serves a purpose.

entries.Add(new Entry(currentDate, prompt, userInput));

and

entry.Display();

both call on this class. Each is called as a method. The constructor (Entry()) is used to set up an instance of that class containing the information. The Display() method allows us to format and display the information contained within the entry. By using a class, we simplify what we need to execute. Instead of running the lines themselves, we instead form a class that not only holds the information but has methods to display it for us. Abstraction can be seen as simplifying code into smaller parts, such as creating classes, that simplify the work that we would need to do in a larger program. Making code easier to repeat saves us from more potential errors, but speeds up the process of finishing our work.