Glen Kelley

10/12/2024

Prove: Articulate – Abstraction

At its core, abstraction is the process of taking concepts that are complicated and contains them in a simpler idea. A simple example of this would be a function that returns the area of a circle. The function accepts a single argument, in this case the radius, and returns a single number, which is the area. Within the function the calculations and any other alterations take place. In this case, the calculations needed to find the circle area is abstracted away and leave only calling the function.

One of the primary benefits of utilizing abstraction while designing a function, class, or program is that this is generally something that can be called upon repeatedly. Such as the function mentioned previously to find the area of a circle. That function can be easily called multiple times in different instances and it would return the area so long as it is give a radius. However, a better example may be the journal project from this week. In this project, a journal and an entry class were created, with one calling the other as a variable type.

public class Journal

{

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

    public void Display()

    {

        foreach(Entry entry in \_entries)

        {

            entry.Display();

        }

    }

}

This would be considered a case of abstraction within abstraction. While the journal class is an abstraction as it contains both the list variable of entries, which is also an abstraction as Entry is another class, it relies on abstraction to work for the Display() function as is only needs to call the Display() function of the Entry class. This allows for a much simplier code than if it had been required to recreate a similar function or class without utilizing the Entry class.

public class Entry

{

    public string \_writer;

    public string \_date;

    public string \_prompt;

    public string \_response;

    public void Display()

    {

        Console.WriteLine($"Writer: {\_writer}");

        Console.WriteLine($"Date: {\_date} - Prompt: {\_prompt}");

        Console.WriteLine($"{\_response}\n");

    }

}