McKenna Johnson

WDD 210 – Programming with Classes

03/14/2025

**Explain the principle of Abstraction**

Abstraction in regards to programming is simple taking complex ideas and simplifying them – ideally by breaking them up into their essential parts. This can be very beneficial when working with large programs, or ones containing many external or differing parts. If you try to stuff a program full of everything in one spot, it will become cluttered and has the potential to break easily (and badly). Abstraction can help separate and simplify the moving pieces – leading to easier fixes, changes, and sharing clarification. (This comes in particularly handy when you need to change one class, and don’t need to change it ten other places as well). An example of this being used is below, pulled from the Journal program I just completed. This section can then be called in the main program as needed with just one of two lines of code.

public *class* Entry

{

    public string \_entryDate;

    public string \_entryPrompt;

    public string \_entryText;

    public string \_goodThing;

    public string \_helpThing;

    public void Display()

    {

        Console.WriteLine($"{\_entryDate} : {\_entryPrompt}");

        Console.WriteLine(\_entryText);

        Console.WriteLine($"One good thing from today: {\_goodThing}");

        Console.WriteLine($"One thing I need help with today: {\_helpThing}");

    }

}