Encapsulation Articulate

Encapsulation is the process of enclosing something or cutting off from other things. In the world of coding, the principle of Encapsulation means to section pieces of your code and take away any unnecessary access to that code. To put it in easy-to-understand terms, if abstraction is like making new rooms to sort and put things away in, encapsulation is setting up locks and keys to the doors to those rooms. Encapsulation benefits a project by controlling the flow of access to each part of the code. If there is ever any problems in your coding, you can quickly find where it is and see what other sections the problem could have spread to. For example, my piece of code here:

using System;

class Reference

{

    private string \_book;

    private int \_chapter;

    private int \_verse;

    private int \_endVerse;

    public string GetDisplayText()

    {

        string text = ();

        return text;

    }

    public Reference(string book, int chapter, int verse)

    {

        \_book = "";

        \_chapter = chapter;

        \_verse = verse;

    }

    public Reference(string book, int chapter, int startVerse, int endVerse)

    {

        \_book = "";

        \_chapter = chapter;

        \_verse = startVerse;

        \_endVerse = endVerse;

    }

    public void Display()

    {

        Console.WriteLine($"{\_book} Chapter {\_chapter}: {\_verse}");

    }

}

The attributes “\_book”, “\_chapter”, “\_verse”, and “\_endverse” are all set to private, meaning that, aside from special codes meant to specifically access them, these lines of code cannot be interacted with by any other parts of the project.