Encapsulation

The concept of encapsulation is to contain each class or portion of a program to itself, enabling them to be independent of one-another and preventing them from sharing details or being able to edit the work of another class unless it is necessary. In the real world, this is similar to people having independent property, with others only being able to use their property with their permission. By encapsulating content, you ensure that each object in the program only has access to the things they need, making each part of the program more succinct, as well as ensuring that different parts of the program don’t accidentally overwrite or break a different part of the program. This also allows different parts of the same program to reuse attribute or variable names without causing a malfunction. This ensures that a program is both more robust and secure while at the same time being more readable.

To share an example, let me display the following code segment:

    private string \_filePath = "ScriptureMastery.csv";

    private bool \_quit = false;

    public Reference \_referenceObject;

In this code segment, the program in questions has several attributes, two of which are marked private while one is marked as public. The two private variables are designed to be used only within this segment of code, and cannot be accessed by other segments. This prevents other program classes (or bad actors) from changing the file path, for example. Meanwhile, the reference object is set as public, so that other classes can continue to work with the attribute. This lets this class build the object with a list of potential references, while the reference class is able to both read this list and make adjustments to it.