Encapsulation is all about hiding certain data in the code in a way that protects it within a class and makes it impossible for that data to be manipulated from other parts of the program. Data that is encapsulated can be manipulated only by its own class and any access to the data comes from the class. The benefit of encapsulation is that when something is broken or needs to be changed, then only the class will need to be looked at, rather than having to search through any other parts of the program that might be affecting the data. In the Scripture Memorizer program created for this class, the Scripture class held the data for both the full Reference and a list of Word objects and had the ability to access and use them:

public class Scripture

{

    private Reference \_reference;

    private List<Word> \_words = new List<Word>();

    public Scripture(Reference reference, string text)

    {

        \_reference = reference;

        string[] wordString = text.Split(" ");

        foreach (string w in wordString)

        {

            Word word = new Word(w);

            \_words.Add(word);

        }

    }

By doing this, the program supplied the strings of words, but the Scripture class controlled how those words went into the list. If the program directly manipulated the list, it would be possible to mess it up, making it dysfunctional. That means that encapsulation helps to keep the program running more smoothly and makes it easier to make needed changes in the future.