Encapsulation means the variables and functions used in a program are contained within classes or objects generated from those classes. They can be made private so that others can’t directly access them or see them. This has the benefit of concealing proprietary code. It also protects the data within the code from being changed in ways that would stop it from working properly; data within a class or object can be made private, and only accessible through methods like getters and setters, so other programmers only interact with it using functions that are specifically written to control how it is used and changed.

Example from this week’s assignment. \_words is an attribute of the Scripture class, and is a list of objects made from the Word class. In this code, instead of setting the \_isHidden attribute of each word I want to hide to “true” directly, I use the Hide() function.

while (i < quantity && remainingWords > 0)

        {

            rand = rng.Next(\_words.Count);

            if (!\_words[rand].IsHidden())

            {

                \_words[rand].Hide();

                i++;

                remainingWords--;

            }

        }