Encapsulation is a fundamental concept in object-oriented programming that involves bundling the data and the methods that operate on the data into a single unit, known as a class. In C#, encapsulation is implemented using access modifiers, which control the visibility and accessibility of the class members. The primary purpose of encapsulation is to protect the data from unauthorized access and modification. By doing this, we can prevent external code from directly accessing and modifying the object’s state, ensuring that the object maintains its integrity. One of the key benefits of encapsulation is improved data security. By restricting direct access to an object's data, we can protect it from accidental or malicious modifications. A practical application of encapsulation can be seen in a banking system, where the details of a bank account need to be protected. For instance, the balance of a bank account should not be directly accessible or modifiable from outside the class. Instead, we provide methods to safely deposit and withdraw money, ensuring that the balance remains consistent and valid.

This is the example of mine code using encapsulation for the variables book, chapter, verse and endVerse inside the class Reference.

public class Reference

{

    private string \_book;

    private int \_chapter;

    private int \_verse;

    private int \_endVerse;

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

    {

        \_book = book;

        \_chapter = chapter;

        \_verse = verse;

    }