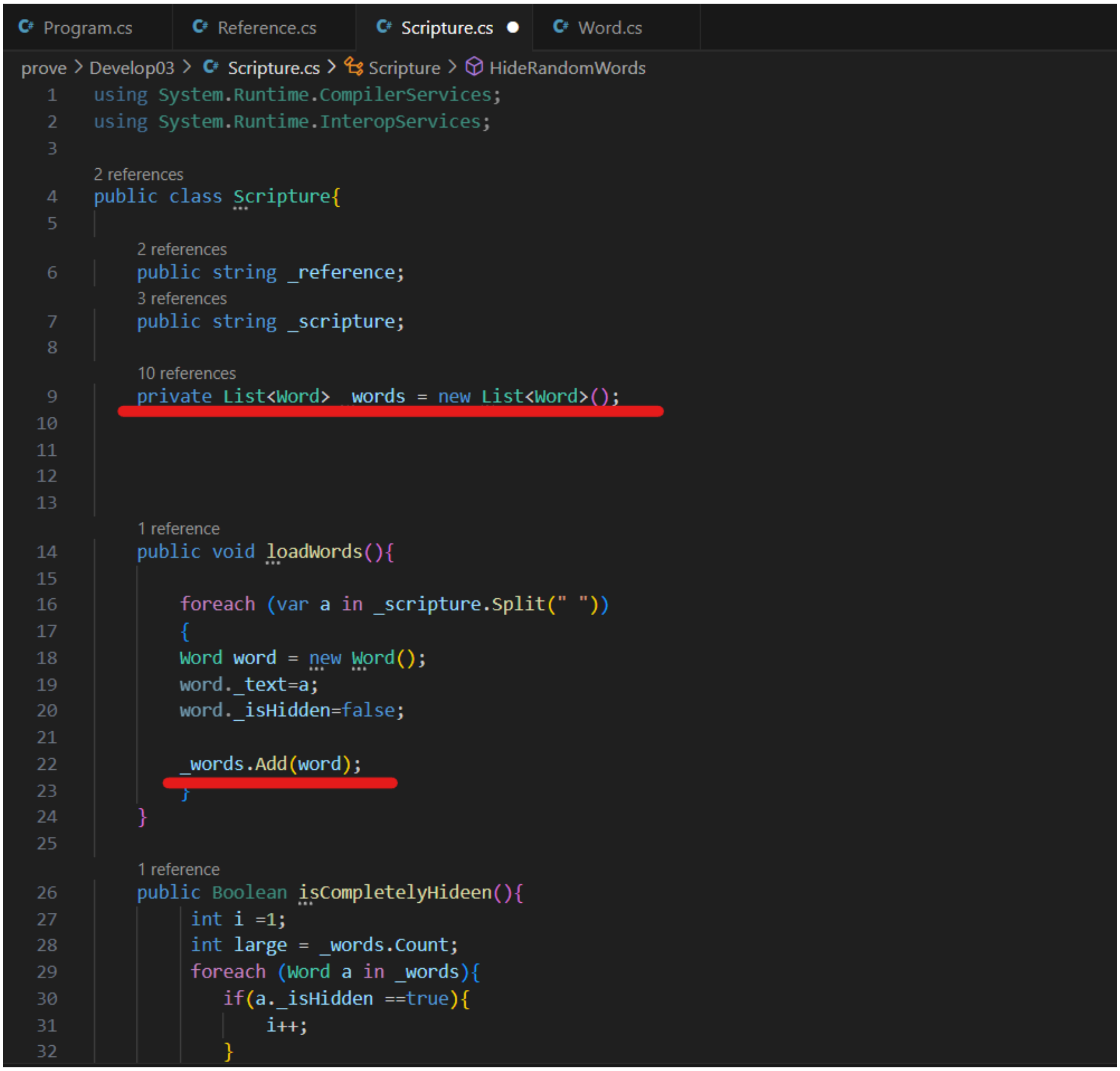
Encapsulation - Articulate

What is encapsulation and why is it important?

- Explain the meaning of Encapsulation
    - The principle of encapsulation basically consists of limiting or closing something so that code cannot be accessed or used from other parts other than only from the class where it was declared and making only the class where it is declared responsible for the details of the specific task, so the containing class has the same logic
- Highlight a benefit of Encapsulation
    - Personally, I consider that one of the highlights of Encapsulation is closing the code, in my own words it would be like adding security to the code and not being used from other parts that could affect the logic and operation of the developed program.
- Provide an application of Encapsulation
    - So that a member of the class can remain encapsulated we must use the word PRIVATE, for example keeping LIST objects private is recommended so that elements cannot be added from other classes other than only in the class to which it belongs, thus we will avoid them from another class adding elements to the list incorrectly or out of logic
- Use a code example of Encapsulation from the program you wrote
    - In may program I used encapsulation in the Scritpture class because in this class I passed the elements of a string to a list, the list was not necessary for other classes to use because all the logic of the work done on the list is contained in the class scripture.cs



```
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

public class Scripture{

    public string _reference;
    public string _scripture;

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




    public void loadWords(){

        foreach (var a in _scripture.Split(" "))
        {
        Word word = new Word();
        word._text=a;
        word._isHidden=false;

        _words.Add(word);
        }
    }

    public Boolean isCompletelyHideen(){
         int i =1;
         int large = _words.Count;
         foreach (Word a in _words){
            if(a._isHidden ==true){
                i++;
            }
```