* **Explain the meaning of Encapsulation.**

Encapsulation is the process of “hiding” or restricting the access to the variables of the code, that means that the variables will be accessed only inside their class, other classes can work with them by accessing through methods. This is to hide the entire process and structure of the class itself and focus on the processes and results.

* **Highlight a benefit of Encapsulation.**

Since you are working with methos (functions) that means that you are able to reuse the same piece of code all the times you need it. You can make changes on the functions without affecting the rest of the code.

* **Provide an application of Encapsulation.**

You can use encapsulation when you have some important information saved in variables that you don’t want other classes to access to. So, you can private the variables and use public methods to interact with them, like some deposits interacting with the balance, etc.

* **Use a code example of Encapsulation from the program you wrote.**
* public string GetDisplayText()
* {
* string wordText="";
* foreach (Word word in \_words)
* {
* wordText += word.GetDisplayText() + " ";
* }
* return wordText;
* }
* **Thoroughly explain these concepts.**

In this function in the class “Scripture” I am using an object or variable type “Word”, that means that this variable is able to access the whole code of the class Word, including the methods. We can’t access directly the variables or attributes of the class Word because we are in a different class; so, what we will do is to use the method GetDisplayText() from the Word class to access the information of the variables that we set before. The GetDisplayText() method accesses variables like Text and make different processes to get the text depending on the hidden words and if the hiding function already hid some words. So, practically, we didn’t use the variables, but we accessed their values with the function. If we change the function, we don’t need to change anything else.