From what I got; the important sections of a program's code can be concealed from other parts that don't need to know about them by using encapsulation. Similar to a secret code, only some portions of the program are permitted to communicate with one another while the rest are kept secret and concealed. It resembles having a locked box. You pack the box with items and lock it, making sure that only you have access to what's inside. The Word class includes a locked box in the memorizer I worked on. Only within the Word class, or the locked box, can the text and hidden variables be accessible.

Encapsulation has the advantage of allowing the object's internal state to be concealed from the outside world, resulting in a clear separation of concerns. This makes it possible to alter the object without changing other portions of the program, and it also makes it simple to find and solve errors.

The "Word" class in this week's program provides an example of how encapsulation is used. Two private attributes, "text" and "hidden," which are only accessible through the class's public methods, are present. While the "IsHidden" function returns the value of the "hidden" attribute, the "Hide" method modifies its value. The attributes are safe and protected from outside interference since they cannot be accessed from outside the class.

Example:

public class Word

{

private string text;

private bool hidden;

public Word(string text)

{

this.text = text;

hidden = false;

}

public bool IsHidden()

{

return hidden;

}

public void Hide()

{

hidden = true;

}

public override string ToString()

{

if (hidden)

{

return "\_\_\_\_\_";

}

else

{

return text;

}

}

}

In the example, the "Word" class's internal state is encapsulated with the "text" and "hidden" attributes being private just like what I have initially done to the memorizer program. The public "Hide" method changes the value of the "hidden" attribute to true, and the public "IsHidden" method returns the value of the "hidden" attribute. The attributes cannot be accessed or modified from outside the class, making them secure and protected from external interference. The "ToString" method uses the "hidden" attribute to either return the hidden word or a series of underscores depending on its value.