**Encapsulation**

**Meaning of Encapsulation**

Encapsulation is a concept in object-oriented programming where we bundle the data and methods that operate on the data into a single unit called a class. By doing this, we control how the data is accessed and changed. We use access modifiers like 'private' to keep some details hidden from the outside world, only exposing what is necessary through 'public' methods.

**Benefit of Encapsulation**

Encapsulation helps in protecting the data from unintended or harmful modifications. It provides a way to enforce rules on how the data should be used, making the code more robust and easier to maintain. By hiding the internal details, we reduce complexity for users of the class, who only need to understand the public interface.

**Application of Encapsulation**

In real-world applications, encapsulation is used to manage and control access to the data within an object. For example, in a banking system, account balances should not be directly accessible or modifiable from outside the class to prevent unauthorized changes. Instead, methods are provided to deposit or withdraw money, ensuring that all operations follow the bank's rules.

**Code Example from my program**

public class Word

{

private string \_text; // This is the data we want to protect

private bool \_isHidden; // Another piece of data to protect

public Word(string text) // Constructor to initialize the word

{

\_text = text;

\_isHidden = false; // By default, the word is not hidden

}

public void Hide() // Method to hide the word

{

\_isHidden = true;

}

public void Show() // Method to show the word

{

\_isHidden = false;

}

public bool IsHidden() // Method to check if the word is hidden

{

return \_isHidden;

}

public string GetDisplayText() // Method to get the word's display text

{

return \_isHidden ? "\_\_\_\_\_" : \_text;

}

}