What is Encapsulation?

Let’s imagine a pill capsule. The outside shell protects the medicine inside. You can't see the medicine or touch it directly. You can only use the whole capsule.

In programming, encapsulation is the same idea. It means putting data in a protected capsule and only letting the outside world interact with it through simple and safe actions.

Why is it important?

It keeps things safe and simple.

- Safe: The data inside is protected. It can't be changed in the wrong way.
- Simple: To use the capsule, you don't need to know how it works on the inside. You just need to know what it does.

An example of code:

We have a Word class and let’s think of it as a capsule for a single word.

- Inside the capsule we have the hidden data:
    - The actual word
    - A secret note saying if the word is hidden or not
- Outside the capsule (the simple actions you can do):
    - Hide() - Tells the capsule to hide the word.
    - GetDisplayText() - Asks the capsule, "Should I show the word or underscores?"

Here's the code for the capsule:

```
public Word(string text)
  {
    _text = text;
    _isHidden = false;
  }

  public void Hide()
  {
```

```
    _isHidden = true;

}

public void Show()
{
    _isHidden = false;

}

public bool IsHidden()
{
    return _isHidden;
}

public string GetDisplayText()
{
    if (_isHidden)
    {
        string hiddenWord = new string('_', _text.Length);
        return hiddenWord;

    }
    else
    {
        return _text;
```

```
}
```

How we use it:

The main program doesn't touch _isHidden directly. It just says: myWord.Hide() and then myWord.GetDisplayText().

This is great because if we want to change how a hidden word looks, we only change the code inside the GetDisplayText method. The rest of the program doesn't need to change at all. The capsule protects the inside and makes updates easy.