Jordan Peterson

Explain Encapsulation Assignment

Encapsulation is a fundamental concept in Object-Oriented Programming that involves keeping the underlining attributes and methods contained privately in a single unit class. It restricts direct access to everything except for methods employing abstraction which are used to interface with other pieces of software.

This is done for several reasons. Sensitive data can be hidden from external access, reducing the risk of accidental modification or corruption, thus providing data protection. By grouping related data and methods together, encapsulation makes code easier to understand, debug, and maintain. It also allows changes to internal logic to be made without affecting other parts of the program that rely on the public interface. This gives you modularity and flexibility.

In the example below, the variables \_text and \_isHidden are set to private. This prevents code from outside the class from interacting with it directly. Outside code must call the hide and show methods. This will prevent problems later should the need arise to change how a word is hidden.

public class Word {

private string \_text;

private bool \_isHidden = false;

public Word(string text)

{

this.\_text = text;

}

public void hide()

{

\_isHidden = true;

}

public void show()

{

\_isHidden = false;

}