1. What is Encapsulation?

Encapsulation is an Object-Oriented Programming (OOP) principle that bundles data (attributes) and behavior (methods) into a single unit (a class) while restricting direct external access to internal details. It enforces data hiding (often using private fields) and provides controlled access via public methods (getters/setters or other operations).

2. Why is Encapsulation Important?

* Prevents unintended modifications: External code can't directly alter internal state, reducing bugs.
* Improves maintainability: Changes to internal logic don’t break dependent code.
* Enhances security: Sensitive data is accessed only through controlled methods.

3. Application in the Scripture Memorization Program

In the program, the Word class encapsulates:

* A \_text field (the word itself).
* A \_isHidden field (tracking visibility).
* Methods like Hide() and GetDisplayText() to safely modify and retrieve data.

4. Code Example

csharp

Copy

public class Word

{

private string \_text;

private bool \_isHidden;

// Constructor initializes the word

public Word(string text)

{

\_text = text;

\_isHidden = false;

}

// Encapsulated method to hide the word

public void Hide()

{

\_isHidden = true;

}

// Controlled access to display text

public string GetDisplayText()

{

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

}

}

5. Explanation of the Example

* Data Protection: \_text and \_isHidden are private, so they can’t be modified directly from outside the class.
* Controlled Behavior: The Hide() method modifies \_isHidden safely, and GetDisplayText() ensures proper formatting when displaying words.
* Flexibility: If we later change how hiding works (e.g., replacing \_\_\_\_\_ with asterisks \*\*\*\*\*), we only modify GetDisplayText()—no other code breaks.

Conclusion

Encapsulation ensures clean, secure, and maintainable code by hiding implementation details and exposing only what’s necessary. In the Scripture program, it lets the Word class manage its own state without interference, making the system more robust and easier to update.