



Poorna Soysa • 2nd
Tech Enthusiast | .NET | C# | Azure | AWS | SAP
1h • 🌐

[Follow](#)

💡 Programming Tip - Design Patterns

🔥 **Design Patterns** are established solutions to recurring problems in software development. Essentially, they are **blueprints** or **templates** that guide developers in creating flexible, reusable, and maintainable code. While not tied to any specific **programming language**, they often align with **object-oriented principles**.

📌 Types of Design Patterns:

Design Patterns are typically categorized into three main groups:

1 Creational Patterns: These patterns deal with object creation mechanisms, trying to create objects flexibly.

Ex: **Singleton, Factory Method, Abstract Factory, Builder, Prototype**

2 Structural Patterns: These patterns deal with how classes and objects are composed to form larger structures.

Ex: **Adapter, Decorator, Bridge, Composite, Proxy, Facade, Flyweight**

3 Behavioural Patterns: These patterns are concerned with the interaction and assignment of responsibilities between objects.

Ex: **Observer, Chain of Responsibility, Command, Interpreter, Strategy, Mediator, Template Method, Memento, Iterator, State, Visitor**

🌟 Benefits of Using Design Patterns:

✅ **Enhanced code clarity:** Patterns provide a shared language among developers, improving code comprehension.

✅ **Enhanced code reusability:** Patterns offer proven solutions that can be applied to different projects.

✅ **Increased flexibility:** Design patterns promote modular and adaptable code.

✅ **Better maintainability:** Well-structured code using patterns is easier to modify and extend.

🚨 **Common Challenges:**

⚠️ **Overuse:** Using patterns for every problem can lead to overly complex code.

⚠️ **Misapplication:** A thorough understanding of a pattern is crucial before implementation.

⚠️ **Adaptability:** Patterns are a starting point and should be adapted to your specific needs.

? How many ***design patterns*** have you used so far? Comment below 📌

📺 Subscribe to my YouTube channel for tutorials, tips, and everything you need to level up your coding skills. ❤️📌

📌 Channel Link: <https://lnkd.in/gER56NV4>

♻️ If this content is useful, ***repost*** to spread the knowledge.

📌 Please follow me ([Poorna Soysa](#)) and click the notification bell icon (🔔) on my profile to receive notifications for all my upcoming posts.

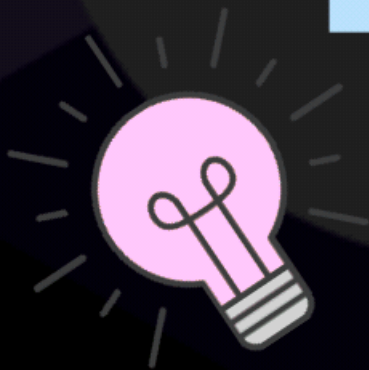
Thank you for reading!

[#DesignPatterns](#) [#DotNET](#) [#DotNETDevelopers](#) [#Programming](#)

Design Patterns

Building Blocks for Better Code

Creational Design Patterns	Structural Design Patterns	Behavioral Design Patterns
Singleton	Adapter	Observer
Factory Method	Decorator	Chain of Responsibility
Abstract Factory	Bridge	Command
Builder	Composite	Interpreter
Prototype	Facade	Iterator
	Flyweight	Mediator
	Proxy	Strategy
		Memento
		State
		Template Method
		Visitor



POORNA SOYSA



REPOST

35

7 comments 6 reposts