ONE PAGE SUMMARY

IT & Software › Design Patterns › Anti Patterns

# Introduction to Anti Patterns

Avoiding Anti Patterns for Better Software Design

## DEFINITION

> Anti patterns are common responses to recurring problems that are ineffective and counterproductive. They highlight poor practices that can lead to negative outcomes in software development.

## KEY CONCEPTS

### Spaghetti Code

A tangled mess of code that is hard to read and maintain, often resulting from poor structure.

### God Object

A single class that knows too much or does too much, leading to high coupling and low cohesion.

### Golden Hammer

Using a familiar tool or approach for every problem, regardless of its suitability.

### Cut-and-Paste Programming

Duplicating code instead of creating reusable components, which increases maintenance difficulty.

### Lava Flow

Code that remains in the system because it is too risky to remove, even if it’s obsolete.

## EXAMPLES

→ Using a single class to handle all database operations, leading to a God Object.

→ Copying and pasting code snippets across multiple files instead of creating a library.

→ Relying on a specific framework for every project, resulting in a Golden Hammer scenario.

## MEMORY TIPS

★ Remember 'SPAGHETTI' for tangled code: S for Spaghetti Code.

★ Think 'GOD' for God Object: it knows everything but is hard to manage.

★ Use 'COPY' for Cut-and-Paste: avoid duplicating code unnecessarily.

## COMMON MISTAKES

✗ Overlooking the importance of code structure, leading to Spaghetti Code.

✗ Assuming one solution fits all, resulting in the Golden Hammer anti pattern.

✗ Neglecting to refactor old code, allowing Lava Flow to accumulate.

## QUICK RECAP

Anti patterns are ineffective solutions to common problems in software design. Recognizing and avoiding them can lead to cleaner, more maintainable code. Focus on best practices to improve your development process.