

The background features a black-to-gray gradient with intricate, white, wavy line patterns that resemble liquid or smoke flowing across the frame. These patterns are most prominent on the left and right sides, framing the central text.

**CREATIONAL DESIGN**

**PATTERNS**

## DEFINITION OF FACTORY PATTERN

The Factory Pattern defines an interface for object creation, allowing subclasses to decide which class to instantiate, promoting loose coupling and flexibility.

## TYPICAL USE CASES

Used in systems producing varied objects like vehicles or notifications, simplifying complex creation logic and enabling runtime decision of instantiated classes.

## KEY PROPERTIES

It creates one product at a time using a single creation method, decoupling client code from concrete classes and localizing subclass instantiation knowledge.

# ABSTRACT FACTORY PATTERN – DEFINITION & PROPERTIES

## Definition of Abstract Factory Pattern

A creational design pattern providing an interface to create families of related objects without specifying their concrete classes, promoting loose coupling.

## Key Properties and Characteristics

Supports multiple creation methods, ensures consistency among product families, encourages platform independence and flexibility.

## Typical Use Cases

Used in GUI frameworks, game development, and systems requiring consistent creation of related objects with high flexibility.

# REAL-WORLD APPLICATIONS, ADVANTAGES & DISADVANTAGES



Real-World Applications of Factory Patterns : Factory Pattern is used in vehicle builders and notification systems.

Advantages of Creational Patterns : These patterns simplify client code, promote flexibility, enhance reusability.

Disadvantages and Limitations : Factory Pattern can become complex with many product types; Abstract Factory adds design complexity and requires understanding multiple components.

# COMPARISON: FACTORY VS ABSTRACT FACTORY

The Factory Pattern creates a single product type via a method like `CreateCar(type)`, promoting loose coupling and simplicity.

Abstract Factory creates families of related products, ensuring consistency and platform independence with higher complexity.

Use Factory for simple single product creation; use Abstract Factory for multiple related products requiring consistency and scalability.

The background is a dark gradient with white, wavy, grid-like lines that create a sense of motion and depth. These lines are concentrated in the corners and edges, leaving the center relatively clear for the text.

**THANK YOU**