# Design Patterns Lec1

Eng/ Mustafa Prince

## Requirements

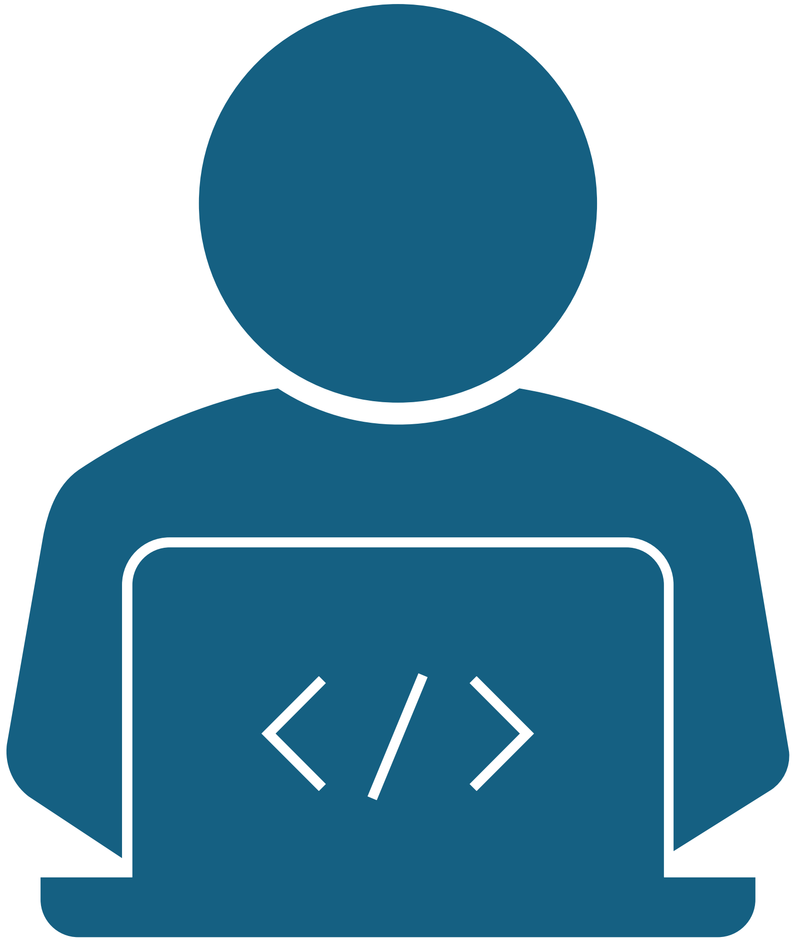
- Any Programming language that support OOP.
- OOP.

Code in this lectures uses dart language.

Sources:

Design Patterns Elements of Reusable Object-Oriented Software book.

Head First Design Patterns book.



## Important Concepts

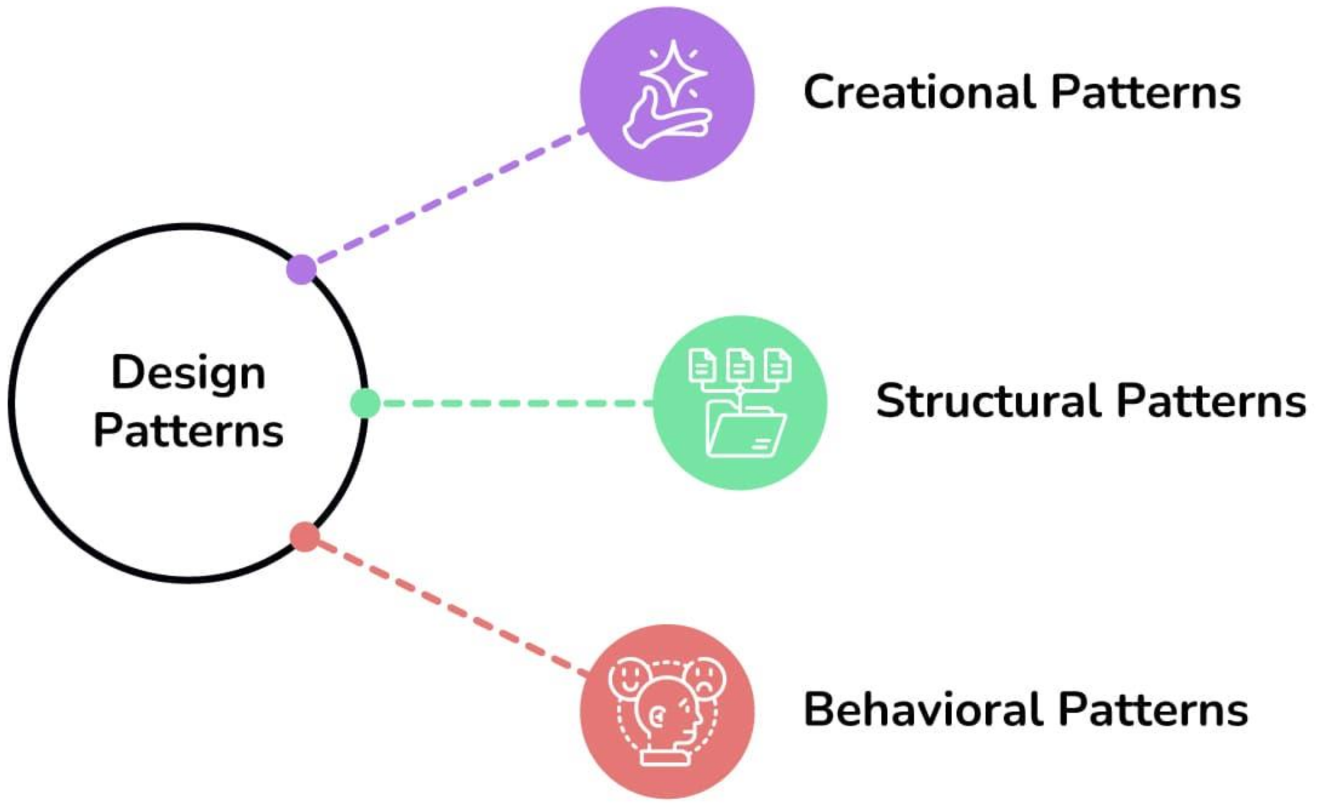
- Architecture Patterns are High-level structures that organize the **overall system design** and responsibilities between layers/modules.
- Solid Principles are Guiding principles (best practices for **code design**).
- Design Patterns are Proven solutions to common problems in **code design.**

## Must we apply design patterns?

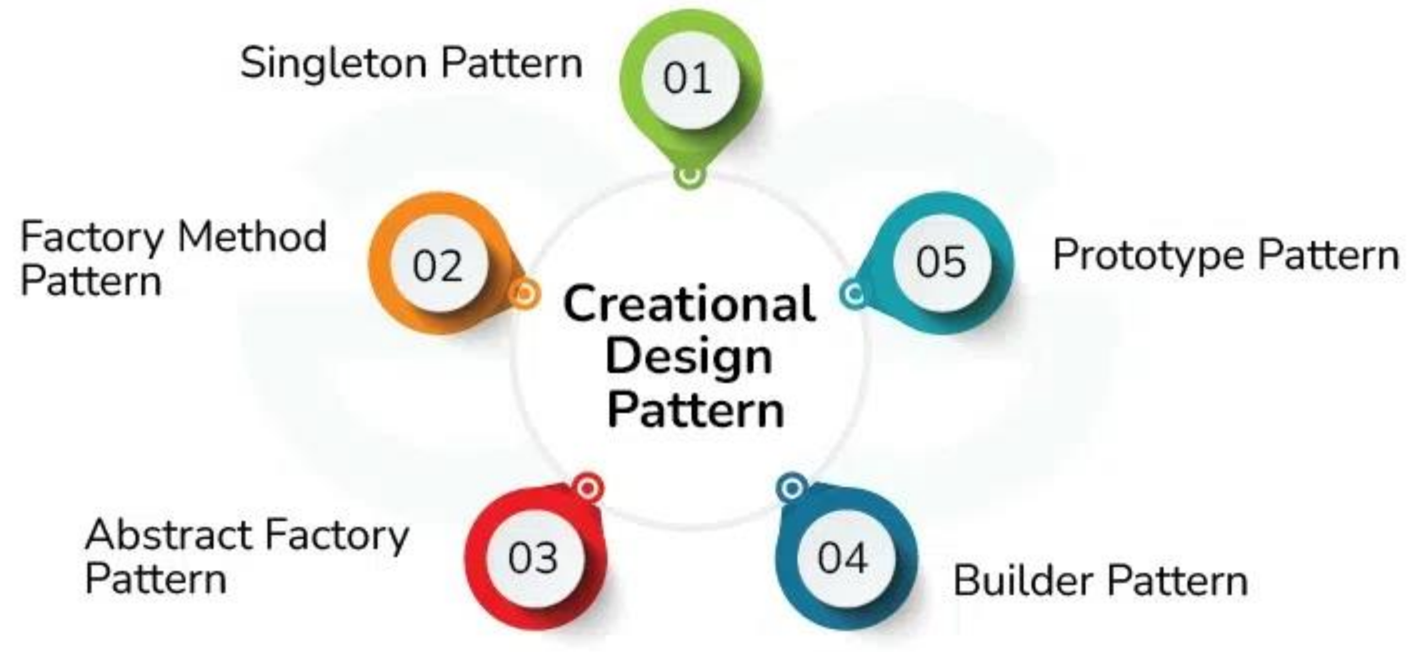
- No, you are not obligated to apply design patterns to every project, but they are highly beneficial for solving common problems, improving code readability, and facilitating communication among developers.

## Design Patterns

- There are three types of design patterns.



# Creational Design Patterns



## Singleton Pattern

- The Singleton Method Design Pattern ensures a class has only one instance and provides a global access point to it.
- In object-oriented programming, the singleton pattern is a software design pattern that restricts the instantiation of a class to a singular instance. It is one of the well-known "Gang of Four" design patterns, which describe how to solve recurring problems in object-oriented software.

## Where I can use singleton

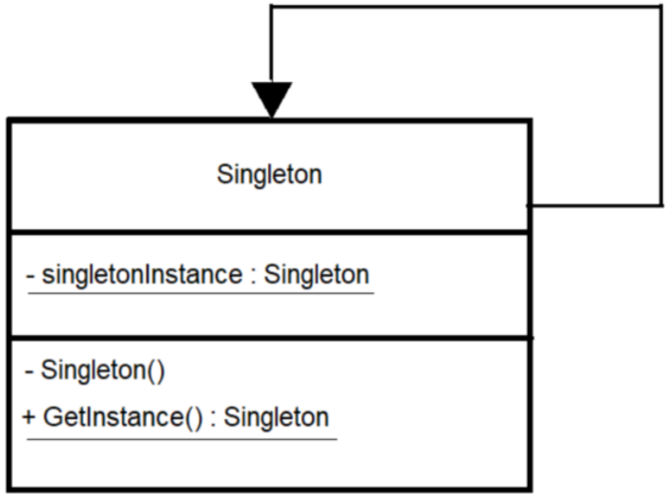
- We need singleton pattern in some applications:
  1. database connection.
  2. logging system.
  3. APIs.

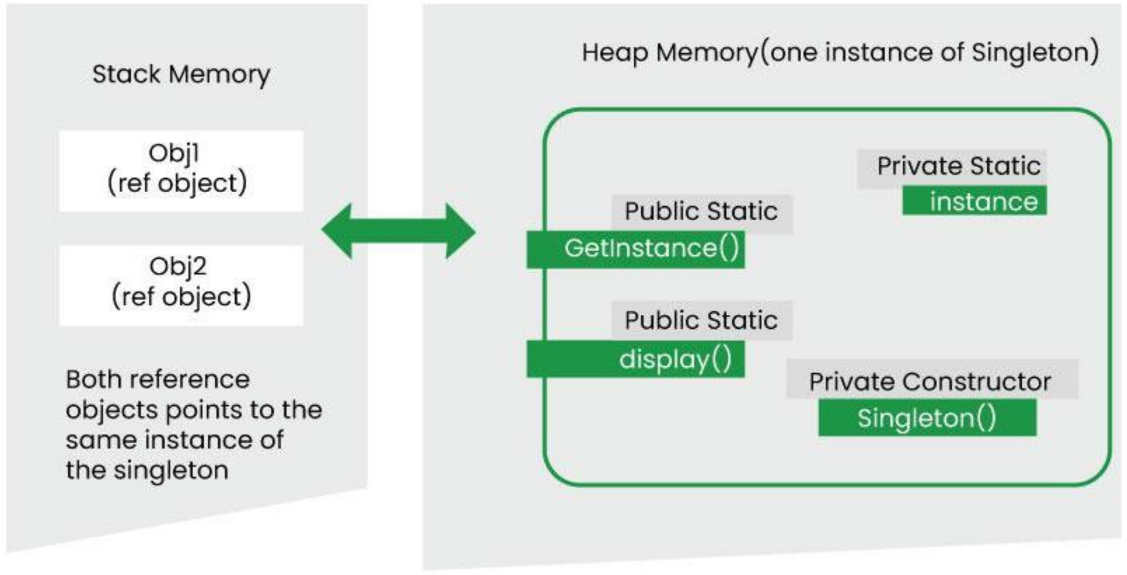
## How to apply this pattern?

1. Make the constructor private.
2. Make static private obj from the class in it.
3. Make static public method to return this instance.

This singleton class is now early initialized.

see how to design a lazy initialized singleton.





# Code Implementation

Thank You