

Design Patterns

This overview of design patterns is taken from the following course : <https://codewithmosh.com/p/design-patterns>

This course provides an overview of a number of design patterns which were originally outlined in the 1994 publication Design Patterns. Elements of Reusable Object-Oriented Software (OOP). The authors and their patterns have come to be known as the Gang of Four : https://en.wikipedia.org/wiki/Design_Patterns

Types of Design patterns

1. *Behavioural* - Focus on the communication between objects.
2. *Structural* - Focus on the strcture of objects and use inheritance to compose interfaces and define ways to compose objects to obtain new functionality.
3. *Creational* - Focus on creating objects, rather than having to instantiate objects directly. This gives the program more flexibility in deciding which objects need to be created for a given case.

The Merits of Design Patterns

The below conversation gives an idea of why Design Patterns are still relevant and will continue to be so :

<https://softwareengineering.stackexchange.com/questions/70877/are-design-patterns-really-essential-nowadays>

Design patterns focus on Best practices taking OOP Design Principles to specific examples, the over all aim is to increase the extensibility, reuseability and maintainability of code. They can be thought of as a a playbook such as that in American Football or Rugby. They aid quick team communcation and provide a range of options for particular scenarios that will help the codes extensibility and reuseabilty. Though they are just scenarios and it is part of the developers job to develop a solution that works best in the given circumstances.