

# Exercises: Design Patterns

This document defines the exercises for ["Java OOP" course @ Software University](#).

## Problem 1. Factory Design Pattern

[https://en.wikipedia.org/wiki/Factory\\_method\\_pattern](https://en.wikipedia.org/wiki/Factory_method_pattern)

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

## Problem 2. Command Design Pattern

[https://en.wikipedia.org/wiki/Command\\_pattern](https://en.wikipedia.org/wiki/Command_pattern)

In Command pattern there is a Command object that encapsulates a request by binding together a set of actions on a specific receiver. It does so by exposing just one method `execute()` that causes some actions to be invoked on the receiver.

## Problem 3. Builder Design Pattern

[https://en.wikipedia.org/wiki/Builder\\_pattern](https://en.wikipedia.org/wiki/Builder_pattern)

It is used to construct a complex object step by step and the final step will return the object. The process of constructing an object should be generic so that it can be used to create different representations of the same object.

## Problem 4. Strategy Design Pattern

[https://en.wikipedia.org/wiki/Strategy\\_pattern](https://en.wikipedia.org/wiki/Strategy_pattern)

Strategy is a behavioral design pattern that lets you define a family of algorithms, put each of them into a separate class, and make their objects interchangeable.