**Creational Patterns**

# Creational Patterns

Creational design patterns are design patterns that deal with object creation mechanisms, trying to create objects in a manner suitable to the situation.

Problem with basic object creation:

* could result in design problems or
* added complexity to the design

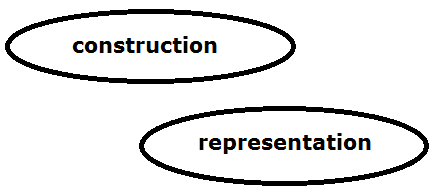
# Abstract Factory

Creates an instance of several families of classes

Provide an interface for creating families of related or dependent objects without specifying their concrete classes.

# Builder

Builder Creational Pattern is used to separate the construction of a complex object from its representation so that the same construction process can create different objects representations.



Builder Creational Pattern

# Factory

A utility class that creates an instance of a class from a family of derived classes.

Define an interface for creating a single object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.

The factory method pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created.

This is done by creating objects by calling a factory method—either specified in an interface and implemented by child classes, or implemented in a base class and optionally overridden by derived classes—rather than by calling a constructor.

**Also known as Virtual Constructor**

# Prototype

# Singleton

Singleton design pattern is used to create exactly one instance of a class.

A global variable makes an object accessible, but it does not keep you from instantiating multiple objects.

A better solution is to make the class itself responsible for keeping track of its sole instance.

The class can ensure that no other instance can be created (by intercepting requests to create new objects), and it can provide a way to access the instance. This is the Singleton pattern.

# References

<https://sourcemaking.com/design_patterns/creational_patterns>

<https://en.wikipedia.org/wiki/Software_design_pattern>