**Design Patterns**

**Introduction:**

Design patterns are generalized solutions to repeated occurring problems in Software Design. We should think of design patterns not as a finished solution but a template to solve a more general problem. There are three main category of design patterns, Creational, Structural and Behavioral design patterns. Each of these categories has a set of specific design patterns that we will examine.

**Creational Design Patterns:**

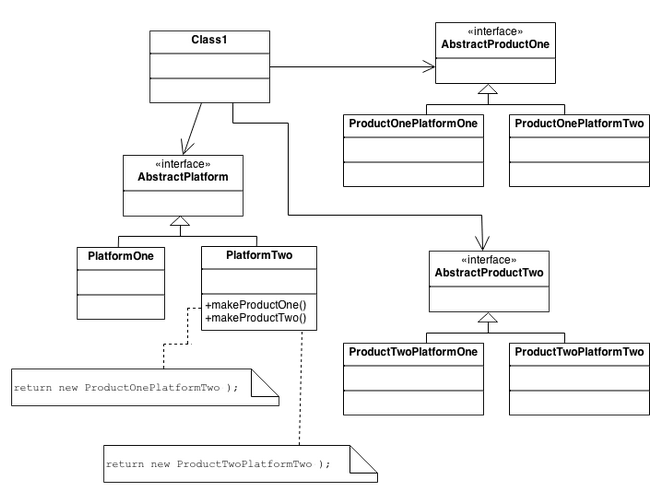
Creational patterns refer to designs that are specific to class instantiation and object creation. In this section we will take a look at the following creational design patterns:

* Abstract Factory
* Builder
* Factory Method
* Object Pool
* Prototype
* Singleton

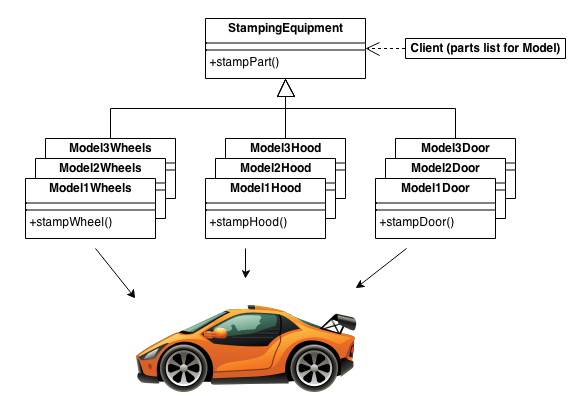
**Abstract Factory:**

The abstract factory design pattern is a pattern intended to provide an interface for creating related objects, without defining their concrete classes. It is typically good to use this design pattern when we are concerned about *portability*, e.g. across *different platform dependencies*, and encapsulating abstracted families of objects.

The abstract factory has factory methods used to create a concrete object part of the family of objects. Figure 1, shows an example of the general structure of an abstract factory design. A more specific example would be of a manufacture that is sheet metal stamping different parts of an automobile. For example, the wheels, the hood, the doors, etc. for different models of cars, which have different layouts of these general parts. We can see the design pattern visually in figure 2.



**Figure 1: An Example of a General Abstract Factory**



**Figure 2: A Specific Use-Case of an Abstract Factory Design Pattern**

**Builder:**

**Structural Design Patterns:**

Structural patterns focus on object composition and interface design. In this section we will take a look at the following structural design patterns:

* Adapter
* Bridge
* Composite
* Decorator
* Façade
* Flyweight
* Private Class Data
* Proxy

**Behavioral Design Patterns:**

Behavior patterns focus mainly on the intercommunication between different objects. In this section we will take a look at the following behavioral design patterns:

* Chain of Responsibility
* Command
* Interpreter
* Iterator
* Mediator
* Memento
* Null Object
* Observer
* State
* Strategy
* Template Method
* Visitor

**References:**

1. <https://sourcemaking.com/design_patterns>
2. Design Patterns (Gang of four: Gamma, Helm, Johnson, Vlissides)
3. Data Structures and Problem Solving Using C++, 2nd-ed, Mark Weiss