**Bridge Design Pattern**

Bridge is a structural design pattern that divides business logic or huge class into separate class hierarchies that can be developed independently.

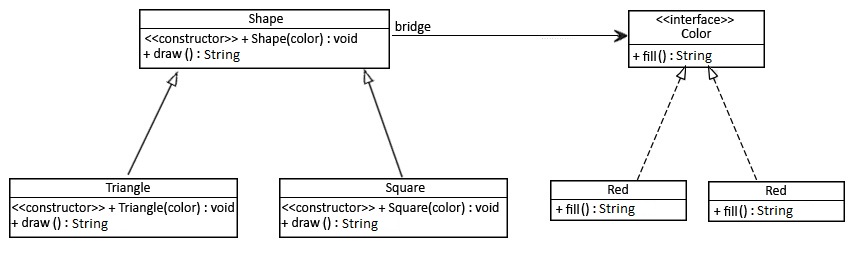
One of these hierarchies (often called the Abstraction) will get a reference to an object of the second hierarchy (Implementation). The abstraction will be able to delegate some (sometimes, most) of its calls to the implementations object. Since all implementations will have a common interface, they’d be interchangeable inside the abstraction.

# **When to Use :**

* When we want a parent abstract class to define the set of basic rules, and the concrete classes to add additional rules
* When we have an abstract class that has a reference to the objects, and it has abstract methods that will be defined in each of the concrete classes

# **Example :**

For the Bridge pattern, we'll consider two layers of abstraction; one is the geometric shape (like triangle and square) which is filled with different colors (our second abstraction layer):

[](https://www.baeldung.com/wp-content/uploads/2017/09/zfq_OUu_M.jpg)