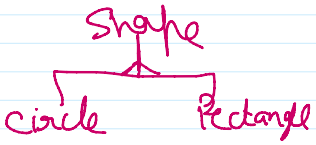
Bridge

17 April 2021 11:35 PM

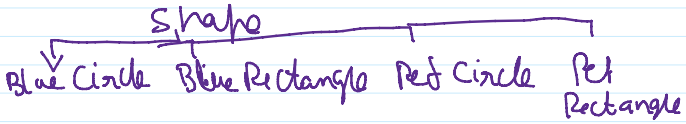
BRIDGE Pattern allows to Decouple an abstraction from Implementation so that they can vary independently.

Eg We have a Shape class




We have Red & Blue Shape so

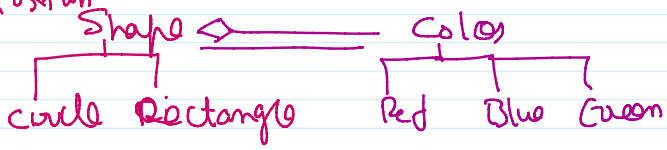
so we need




Total 4

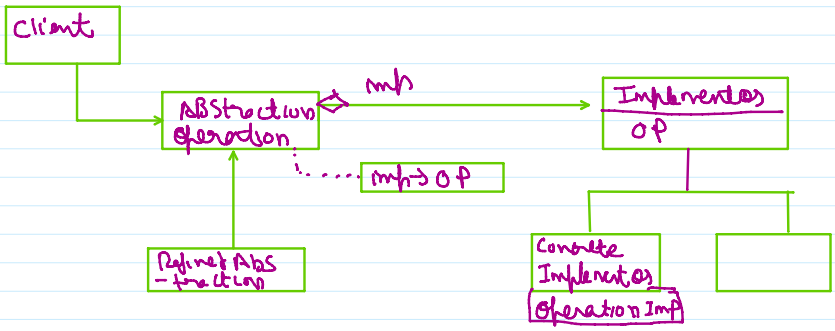
So Bridge Pattern allows to decouple like this

using composition



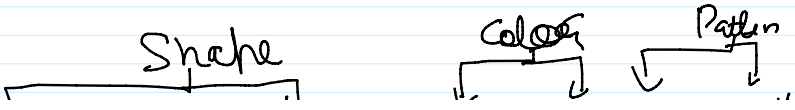

## Applicability

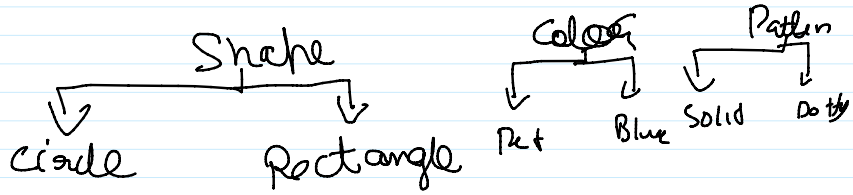
- Wants to avoid permanent binding b/w an abstraction & its implementation.
- Changes in the implementation should have no impact on clients.

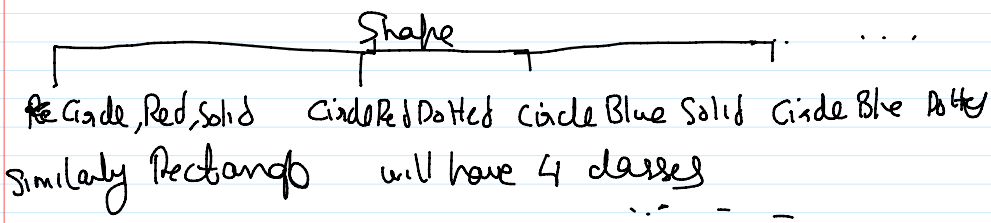



## Participants

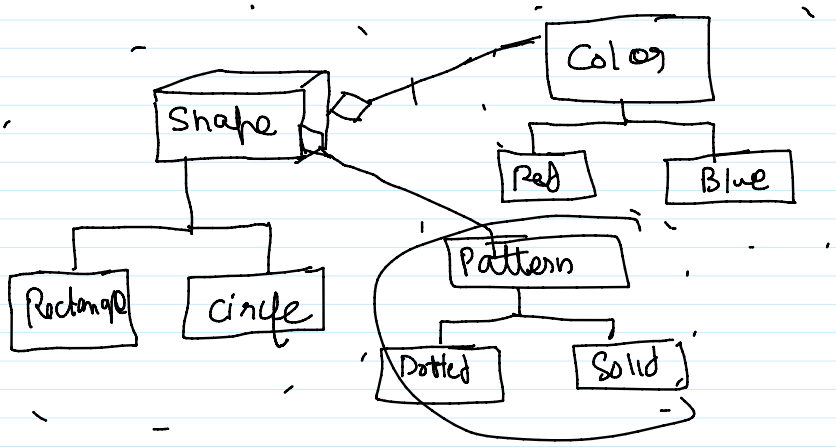
- Abstraction
  - defines the abstraction Interface
  - Maintains a reference to an object of Implementor
- Refined Abstraction
  Extends the abstraction Interface
- Implementor
  - Defines the interface for implementation classes.
- Concrete Implementor
  Extends Implementor.






Similarly Rectangle will have 4 classes




client wants

Red Colored Rectangle with Solid

Grey

2D inheritance