

<b>Abstract Factory</b>	<b>Adapter</b>	<b>Bridge</b>	<b>Composite</b>	<b>Factory Method</b>		
Implement abstraction and interfaces to know what are you working with	How to implement a converter using abstraction and interfaces	How to segregate your interfaces and abstraction therefore it's easier to make modifications	How to implement your abstraction and interfaces like a Tree model	Decoupled your codes by implementing interfaces and abstraction		
<b>Decorator</b>	<b>Singleton</b>	<b>Builder</b>	<b>Prototype</b>	<b>Facade</b>	<b>Flyweight</b>	<b>Proxy</b>
How to make changes to your class by wrapping them inside a decorator	How to have a global but one instance of objects	Reduce your constructor parameter using multiple functions inside the class on the same instance	Instead of letting cloning happen outside the class bring it inside using a method like clone and then return a fresh copy of the class	Create a class to handle all the methods that are required to be run	Memoization using classes and abstraction	Use abstraction and interfaces as a middleware to control access
<b>Command</b>	<b>Chain of Responsibility</b>	<b>Iterator</b>	<b>Mediator</b>	<b>Observer</b>	<b>Memento</b>	<b>Strategy</b>
<b>State</b>	<b>Visitor</b>	<b>Template method</b>				