Decorator Design Pattern

- Decorator design pattern is structural design pattern.
- This pattern acts as wrapper to existing object of class.
- This pattern allows to add new functionality to existing object of class dynamically.
- This doesn’t change the instance or functionality of other class instances, just new objects get new behaviour.
- It does not change the base code class and its object.
- This pattern creates decorator classes which wraps the original class object without changing the original class method signature.

- Implementation
    - This pattern is based on abstract classes and we derive concrete implementation from those classes.
    - In this pattern we create interface that creates blueprint of class which will have decorator functionality and classes.

- Components of decorator pattern
    - Component: interface
        - Its an interface that is used to add functionality to classes at run time
    - Concrete component
        - Its an object to which functionality can be added at runtime.
    - Decorator
        - It refers to component interface and is an interface itself that conforms component interface
    - Concrete Decorator
        - It adds responsibility to components