

Answers:

Q1. Decorator Pattern :

This pattern will be best suited for this problem because we need the capabilities of inheritance but at the same time we also need to add functionalities at the run time

Q2. Observer Pattern :

This pattern will be best suited here because we need to notify the observers of the change or updation in events rather than the observer sending the request again and again to view the changes. It also displays a one to many relationship where many objects depends on one object. When state of one object changes , its dependent object needs to be notified.

Q3. Singleton Pattern

This pattern will be best suited for this problem because we have a requirement that only one instance of the class should be created and all the threads should be working on the same value.

Q4. Strategy Pattern

This pattern will be best suited here because we have multiple algorithms for a specific task and the client decides the actual implementation to be used at the runtime. And the algorithms vary independently from the clients that wish to use it