

# 27 Lecture - CS504

## Important Mcqs

**Q: What is the Observer Pattern in software design?** a) A pattern for creating new objects. b) A pattern for optimizing code performance. c) A behavioral pattern for real-time communication between objects. d) A pattern for handling exceptions in code. **Solution: c) A behavioral pattern for real-time communication between objects.**

**Q: What are the main components in the Observer Pattern?** a) Subject and Listener. b) Observable and Subscriber. c) Observer and Subscriber. d) Subject and Observer. **Solution: d) Subject and Observer.**

**Q: In the Observer Pattern, what is the role of the Subject?** a) It listens to changes in the Observer. b) It updates the Observer with new data. c) It maintains a list of Observers and notifies them of state changes. d) It observes multiple Observers simultaneously. **Solution: c) It maintains a list of Observers and notifies them of state changes.**

**Q: What is the benefit of using the Observer Pattern?** a) Improved code performance. b) Reduced code modularity. c) Enhanced code reusability. d) Increased code complexity. **Solution: c) Enhanced code reusability.**

**Q: Which design principle does the Observer Pattern adhere to?** a) Liskov Substitution Principle. b) Open/Closed Principle. c) Single Responsibility Principle. d) Dependency Inversion Principle. **Solution: d) Dependency Inversion Principle.**

**Q: In the Observer Pattern, what happens when the Subject's state changes?** a) The Subject updates its state to match the Observer's state. b) The Observer updates its state to match the Subject's state. c) The Subject notifies all registered Observers, and they update themselves. d) The Observer triggers the state change in the Subject. **Solution: c) The Subject notifies all registered Observers, and they update themselves.**

**Q: Which pattern promotes loose coupling between the Subject and Observers?** a) Adapter Pattern. b) Strategy Pattern. c) Observer Pattern. d) Singleton Pattern. **Solution: c) Observer Pattern.**

**Q: How does the Observer Pattern support real-time communication?** a) By using blocking I/O operations. b) By using synchronous method calls. c) By maintaining a list of Observers and notifying them of state changes. d) By using multithreading. **Solution: c) By maintaining a list of Observers and notifying them of state changes.**

**Q: What happens if a new Observer is added to the Subject after a state change?** a) The new Observer will be notified of the previous state change. b) The new Observer will be notified only of future state changes. c) The new Observer will update the Subject's state. d) The new Observer will not receive any notifications. **Solution: b) The new Observer will be notified only of future state changes.**

**Q: Which design pattern is commonly used to implement event handling in graphical user interfaces (GUI)?** a) Factory Method Pattern. b) Observer Pattern. c) Singleton Pattern. d) Decorator Pattern. **Solution: b) Observer Pattern.**