Strategy Pattern

1. **Which pattern allows a family of algorithms to be interchangeable within a single class?**
   - A) Observer Pattern
   - B) Strategy Pattern
   - C) Mediator Pattern
   - D) Command Pattern
   - **Answer:** B) Strategy Pattern

2. **In the Strategy Pattern, how is the specific algorithm chosen?**
   - A) At compile time
   - B) At runtime
   - C) At design time
   - D) After the client object is destroyed
   - **Answer:** B) At runtime

3. **Which of the following best describes the Strategy Pattern?**
   - A) Encapsulates each algorithm in a separate class
   - B) Ties the algorithm to the client class
   - C) Limits the number of algorithms that can be used
   - D) Requires all algorithms to be implemented in the same class
   - **Answer:** A) Encapsulates each algorithm in a separate class

4. **Which method in the `SortContext` class triggers the execution of the selected strategy?**
   - A) `ExecuteStrategy()`
   - B) `SortList()`
   - C) `ApplyStrategy()`
   - D) `RunStrategy()`
   - **Answer:** B) SortList()

5. **In the Strategy Pattern, what does the client object hold a reference to?**

- A) A specific algorithm implementation
- B) An abstract strategy interface
- C) A mediator object
- D) A command object
- **Answer:** B) An abstract strategy interface

6. **What would be a disadvantage of using the Strategy Pattern?**
- A) Increases the number of classes
- B) Tightly couples the context to a specific algorithm
- C) Reduces code flexibility
- D) Makes algorithms impossible to change at runtime
- **Answer:** A) Increases the number of classes

7. **In the context of the Strategy Pattern, what role does `SortContext` play?**
- A) Concrete strategy
- B) Context
- C) Strategy interface
- D) Observer
- **Answer:** B) Context

8. **Why might you choose to use the Strategy Pattern in a program?**
- A) To reduce the number of classes
- B) To encapsulate varying behavior in different classes
- C) To enforce a single algorithm across all clients
- D) To avoid creating interfaces
- **Answer:** B) To encapsulate varying behavior in different classes

9. **If a new sorting algorithm needs to be added, what must be done when using the Strategy Pattern?**
- A) Modify the existing `SortContext` class
- B) Add a new class that implements `ISortStrategy`

- C) Change the base `ISortStrategy` interface
- D) Modify all existing strategy classes
- **Answer:** B) Add a new class that implements `ISortStrategy`

10. **In the Strategy Pattern, who decides which strategy to use?**
    - A) The client
    - B) The strategy interface
    - C) The context class
    - D) The mediator
    - **Answer:** A) The client

Observer Pattern

11. **Which pattern defines a one-to-many dependency between objects?**
    - A) Command Pattern
    - B) Observer Pattern
    - C) Strategy Pattern
    - D) Mediator Pattern
    - **Answer:** B) Observer Pattern

12. **In the Observer Pattern, what happens when the subject changes state?**
    - A) All observers are automatically notified and updated
    - B) Only the first observer is notified
    - C) Observers must poll the subject to get updates
    - D) The subject remains unchanged
    - **Answer:** A) All observers are automatically notified and updated

13. **What is the main role of the `Notify()` method in the Observer Pattern?**
    - A) To change the state of the subject
    - B) To remove an observer
    - C) To alert all registered observers of a state change

- D) To initialize the observers
- **Answer:** C) To alert all registered observers of a state change

14. **Which of the following is a key characteristic of the Observer Pattern?**
    - A) Tight coupling between subject and observers
    - B) Loose coupling between subject and observers
    - C) Strong dependency on a specific observer
    - D) None of the above
    - **Answer:** B) Loose coupling between subject and observers

15. **What is an advantage of using the Observer Pattern?**
    - A) Simplifies the design by avoiding the use of interfaces
    - B) Ensures that all observers are tightly coupled with the subject
    - C) Allows dynamic addition or removal of observers at runtime
    - D) Prevents any observer from being notified more than once
    - **Answer:** C) Allows dynamic addition or removal of observers at runtime

16. **Which method in the `Stock` class is responsible for updating observers?**
    - A) `Detach()`
    - B) `Attach()`
    - C) `Notify()`
    - D) `SetPrice()`
    - **Answer:** C) Notify()

17. **In the Observer Pattern, what type of relationship exists between the subject and its observers?**
    - A) One-to-one
    - B) One-to-many
    - C) Many-to-one
    - D) Many-to-many
    - **Answer:** B) One-to-many

18. **Which design issue does the Observer Pattern address?**
    - A) How to encapsulate different algorithms
    - B) How to notify multiple objects when a single object changes state
    - C) How to manage a chain of request handlers
    - D) How to mediate communication between objects
    - **Answer:** B) How to notify multiple objects when a single object changes state

19. **How do observers register with a subject in the Observer Pattern?**
    - A) By implementing a shared interface
    - B) By directly accessing the subject's internal state
    - C) By calling the subject's `Attach()` method
    - D) By being declared as a friend class of the subject
    - **Answer:** C) By calling the subject's `Attach()` method

20. **What is a potential downside of the Observer Pattern?**
    - A) Increased complexity due to multiple observers
    - B) Difficulty in dynamically adding observers
    - C) Tight coupling between observers and the subject
    - D) Limited scalability due to a fixed number of observers
    - **Answer:** A) Increased complexity due to multiple observers

Command Pattern

21. **Which pattern encapsulates a request as an object?**
    - A) Strategy Pattern
    - B) Command Pattern
    - C) Observer Pattern
    - D) Mediator Pattern
    - **Answer:** B) Command Pattern

22. **In the Command Pattern, what role does the `LightOnCommand` class play?**

- A) Receiver
- B) Command
- C) Invoker
- D) Client
- **Answer:** B) Command

23. **Which of the following is NOT a benefit of the Command Pattern?**

- A) Queuing requests for execution
- B) Logging changes for possible undo
- C) Eliminating the need for a receiver class
- D) Supporting batch commands
- **Answer:** C) Eliminating the need for a receiver class

24. **Which method in the Command Pattern is responsible for carrying out the command?**

- A) `Execute()`
- B) `Perform()`
- C) `Run()`
- D) `Invoke()`
- **Answer:** A) Execute()

25. **In the Command Pattern, which component knows how to execute the command?**

- A) The client
- B) The invoker
- C) The command itself
- D) The strategy
- **Answer:** C) The command itself

26. **Which of the following best describes the relationship between the invoker and the command in the Command Pattern?**

- A) The invoker is responsible for creating the command

- B) The invoker holds a reference to the command and calls its `Execute()` method
- C) The command directly invokes methods on the invoker
- D) The invoker and command are the same object
- **Answer:** B) The invoker holds a reference to the command and calls its `Execute()` method

27. **In the Command Pattern, what is the role of the receiver?**
- A) To trigger the command execution
- B) To perform the actual work requested by the command
- C) To create the command object
- D) To decide which command to execute
- **Answer:** B) To perform the actual work requested by the command

28. **What would be a typical use case for the Command Pattern?**
- A) Creating complex user interfaces
- B) Implementing undo/redo functionality
- C) Managing a set of observers
- D) Dynamically changing algorithms
- **Answer:** B) Implementing undo/redo functionality

29. **In the example provided,

what does the `RemoteControl` class represent?**
- A) Receiver
- B) Command
- C) Invoker
- D) Client
- **Answer:** C) Invoker

30. **Which statement about the Command Pattern is FALSE?**
- A) It allows requests to be queued and executed later
- B) It can be used to support macro commands

- C) It reduces the number of classes in a system
- D) It allows logging of operations for potential undo functionality
- **Answer:** C) It reduces the number of classes in a system

Chain of Responsibility Pattern

31. **Which pattern allows a request to be passed along a chain of handlers?**
- A) Observer Pattern
- B) Command Pattern
- C) Chain of Responsibility Pattern
- D) Mediator Pattern
- **Answer:** C) Chain of Responsibility Pattern

32. **In the Chain of Responsibility Pattern, what happens if a handler cannot process a request?**
- A) The request is discarded
- B) The request is passed to the next handler in the chain
- C) The request is processed with an error
- D) The handler retries the request
- **Answer:** B) The request is passed to the next handler in the chain

33. **Which method is typically used to pass the request along the chain in the Chain of Responsibility Pattern?**
- A) `Execute()`
- B) `HandleRequest()`
- C) `PassRequest()`
- D) `ProcessRequest()`
- **Answer:** B) HandleRequest()

34. **What is the primary advantage of the Chain of Responsibility Pattern?**
- A) All requests are guaranteed to be handled
- B) It allows multiple objects to handle the request in sequence

- C) It ensures requests are handled in a specific order
- D) It limits the number of handlers that can process a request
- **Answer:** B) It allows multiple objects to handle the request in sequence

35. **Which of the following is NOT a characteristic of the Chain of Responsibility Pattern?**
- A) Decoupling of sender and receiver
- B) Multiple handlers for a single request
- C) A single, fixed handler for each request
- D) Dynamic determination of the request handler at runtime
- **Answer:** C) A single, fixed handler for each request

36. **In the example provided, what does the `LevelOneSupport` class represent?**
- A) Client
- B) Handler
- C) Request
- D) Invoker
- **Answer:** B) Handler

37. **How does a handler in the Chain of Responsibility Pattern determine if it should handle a request?**
- A) By checking its position in the chain
- B) By examining the content of the request
- C) By consulting the invoker
- D) By using a strategy pattern
- **Answer:** B) By examining the content of the request

38. **Which of the following scenarios is a good fit for the Chain of Responsibility Pattern?**
- A) Implementing a menu system in a UI
- B) Validating input through a sequence of checks
- C) Managing a list of event listeners
- D) Dynamically choosing a sorting algorithm

- **Answer:** B) Validating input through a sequence of checks

39. **What would happen if none of the handlers in the chain can process the request?**
    - A) The request is handled by a default handler
    - B) The request is discarded or an error is returned
    - C) The request is automatically handled by the first handler
    - D) The request is sent back to the client
    - **Answer:** B) The request is discarded or an error is returned

40. **Which statement is true about the Chain of Responsibility Pattern?**
    - A) Each request must be handled by all handlers in the chain
    - B) The order of handlers in the chain does not matter
    - C) Handlers can be added or removed from the chain dynamically
    - D) A request is always handled by the first handler in the chain
    - **Answer:** C) Handlers can be added or removed from the chain dynamically

Mediator Pattern

41. **Which pattern defines an object that encapsulates how a set of objects interact?**
    - A) Command Pattern
    - B) Observer Pattern
    - C) Mediator Pattern
    - D) Strategy Pattern
    - **Answer:** C) Mediator Pattern

42. **In the Mediator Pattern, what role does the `ChatRoom` class play?**
    - A) Mediator
    - B) Concrete Colleague
    - C) Command
    - D) Invoker
    - **Answer:** A) Mediator

43. **Which of the following best describes the Mediator Pattern?**
    - A) It eliminates direct communication between objects
    - B) It enforces direct communication between objects
    - C) It allows objects to communicate without a central mediator
    - D) It promotes tightly coupled object interactions
    - **Answer:** A) It eliminates direct communication between objects

44. **What is a primary advantage of using the Mediator Pattern?**
    - A) Reduces the number of communication paths between objects
    - B) Increases the number of classes in the system
    - C) Enforces strict one-to-one communication
    - D) Requires all communication to go through multiple intermediaries
    - **Answer:** A) Reduces the number of communication paths between objects

45. **In the example provided, what role does the `User` class play?**
    - A) Mediator
    - B) Concrete Colleague
    - C) Invoker
    - D) Command
    - **Answer:** B) Concrete Colleague

46. **Which of the following is a potential downside of the Mediator Pattern?**
    - A) Increased complexity due to more communication paths
    - B) The mediator can become a performance bottleneck
    - C) Direct communication between objects becomes difficult to achieve
    - D) The mediator must handle all types of communication
    - **Answer:** B) The mediator can become a performance bottleneck

47. **When might you choose to use the Mediator Pattern?**
    - A) When objects are highly dependent on each other

- B) When you want to reduce the number of dependencies between objects
- C) When every object should communicate directly with every other object
- D) When you want to remove the need for a central coordinating object
- **Answer:** B) When you want to reduce the number of dependencies between objects

48. **Which statement about the Mediator Pattern is TRUE?**
- A) It promotes tight coupling between objects
- B) It replaces the need for all communication between objects
- C) It centralizes complex communication logic
- D) It requires that each object knows about every other object
- **Answer:** C) It centralizes complex communication logic

49. **How does the Mediator Pattern affect the maintainability of code?**
- A) It decreases maintainability by increasing the number of classes
- B) It increases maintainability by reducing direct dependencies
- C) It decreases maintainability by promoting tight coupling
- D) It has no effect on maintainability
- **Answer:** B) It increases maintainability by reducing direct dependencies

50. **In the context of the Mediator Pattern, what is a `Colleague`?**
- A) The object that communicates directly with others
- B) An object that interacts with other colleagues through the mediator
- C) The central object that handles communication
- D) A helper object used by the mediator
- **Answer:** B) An object that interacts with other colleagues through the mediator

Mixed Behavioral Patterns

51. **Which of the following patterns promotes loose coupling between sender and receiver?**
- A) Strategy Pattern
- B) Chain of Responsibility Pattern

- C) Observer Pattern
- D) Command Pattern
- **Answer:** C) Observer Pattern

52. **Which pattern is most likely to be used when implementing an undo/redo feature?**
- A) Strategy Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** C) Command Pattern

53. **Which pattern is best suited for managing event propagation in a GUI framework?**
- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Mediator Pattern
- D) Strategy Pattern
- **Answer:** B) Chain of Responsibility Pattern

54. **In which pattern would you most likely find a `ConcreteStrategy` class?**
- A) Command Pattern
- B) Observer Pattern
- C) Mediator Pattern
- D) Strategy Pattern
- **Answer:** D) Strategy Pattern

55. **Which of the following patterns is primarily concerned with handling a sequence of operations or commands?**
- A) Chain of Responsibility Pattern
- B) Mediator Pattern
- C) Command Pattern
- D) Observer Pattern

- **Answer:** C) Command Pattern

56. **Which pattern is most appropriate for broadcasting a change in state to multiple objects?**
    - A) Strategy Pattern
    - B) Observer Pattern
    - C) Mediator Pattern
    - D) Command Pattern
    - **Answer:** B) Observer Pattern

57. **Which pattern can be used to dynamically change the behavior of an object at runtime?**
    - A) Chain of Responsibility Pattern
    - B) Command Pattern
    - C) Strategy Pattern
    - D) Mediator Pattern
    - **Answer:** C) Strategy Pattern

58. **Which pattern would be the best choice for decoupling the sender and receiver of a request?**
    - A) Observer Pattern
    - B) Chain of Responsibility Pattern
    - C) Command Pattern
    - D) Strategy Pattern
    - **Answer:** C) Command Pattern

59. **Which pattern is typically used to centralize complex communication between multiple objects?**
    - A) Observer Pattern
    - B) Chain of Responsibility Pattern
    - C) Mediator Pattern
    - D) Command Pattern
    - **Answer:** C) Mediator Pattern

60. **Which pattern is characterized by a chain of handlers that process requests?**

- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** B) Chain of Responsibility Pattern

Advanced Behavioral Pattern Questions

61. **In the Strategy Pattern, how can you change the algorithm being used by an object?**

- A) By changing the concrete strategy class at runtime
- B) By modifying the context class
- C) By altering the client code
- D) By replacing the observer
- **Answer:** A) By changing the concrete strategy class at runtime

62. **What is a key difference between the Command Pattern and the Strategy Pattern?**

- A) The Strategy Pattern deals with encapsulating algorithms, while the Command Pattern encapsulates requests.
- B) The Command Pattern supports undo operations, while the Strategy Pattern does not.
- C) The Strategy Pattern is used for object creation, while the Command Pattern is used for object behavior.
- D) The Command Pattern promotes loose coupling, while the Strategy Pattern does not.
- **Answer:** A) The Strategy Pattern deals with encapsulating algorithms, while the Command Pattern encapsulates requests.

63. **In the Observer Pattern, what is the primary responsibility of the subject?**

- A) To notify all attached observers when its state changes
- B) To handle requests passed to it
- C) To encapsulate different algorithms
- D) To mediate communication between observers
- **Answer:** A) To notify all attached observers when its state changes

64. **Which design pattern would you use to manage multiple event listeners in an application?**

- A) Command Pattern
- B) Observer Pattern
- C) Chain of Responsibility Pattern
- D) Mediator Pattern
- **Answer:** B) Observer Pattern

65. **Which pattern is most suitable for processing a request that can be handled by more than one handler in sequence?**

- A) Strategy Pattern
- B) Chain of Responsibility Pattern
- C) Observer Pattern
- D) Command Pattern
- **Answer:** B) Chain of Responsibility Pattern

66. **Which pattern would be best for handling multiple actions triggered by a single user input?**

- A) Strategy Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** C) Command Pattern

67. **What is the primary role of the `Receiver` in the Command Pattern?**

- A) To encapsulate the request
- B) To execute the request when the command is invoked
- C) To determine which command should be executed
- D) To observe the state of the command
- **Answer:** B) To execute the request when the command is invoked

68. **Which pattern allows objects to communicate without knowing each other's identities?**

- A) Command Pattern
- B) Chain of Responsibility Pattern
- C) Observer Pattern
- D) Mediator Pattern
- **Answer:** D) Mediator Pattern

69. **In the Chain of Responsibility Pattern, how is the order of handlers typically determined?**
- A) By the order in which handlers are added to the chain
- B) By a priority assigned to each handler
- C) Randomly at runtime
- D) By the invoker object
- **Answer:** A) By the order in which handlers are added to the chain

70. **What is a potential drawback of the Chain of Responsibility Pattern?**
- A) It creates a single point of failure
- B) The request may go unhandled if no suitable handler is found
- C) It tightly couples the request to a specific handler
- D) All handlers must process the request
- **Answer:** B) The request may go unhandled if no suitable handler is found

Real-World Behavioral Pattern Scenarios

71. **Which pattern would you use to implement a customer support ticket system where tickets escalate to different support levels?**
- A) Command Pattern
- B) Observer Pattern
- C) Chain of Responsibility Pattern
- D) Strategy Pattern
- **Answer:** C) Chain of Responsibility Pattern

72. **In a home automation system, which pattern would be best for issuing commands to devices like lights and thermostats?**

- A) Observer Pattern
- B) Strategy Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** C) Command Pattern

73. **Which pattern would be suitable for a stock market system where investors need to be notified of price changes?**

- A) Command Pattern
- B) Observer Pattern
- C) Chain of Responsibility Pattern
- D) Mediator Pattern
- **Answer:** B) Observer Pattern

74. **Which pattern would you use to implement a chat application where users communicate through a central server?**

- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Mediator Pattern
- D) Strategy Pattern
- **Answer:** C) Mediator Pattern

75. **Which pattern would be most appropriate for managing multiple validation checks on user input in a form?**

- A) Strategy Pattern
- B) Chain of Responsibility Pattern
- C) Observer Pattern
- D) Command Pattern
- **Answer:** B) Chain of Responsibility Pattern

76. **Which pattern would you use to allow users to select different sorting algorithms in an application?**

- A) Command Pattern
- B) Observer Pattern
- C) Strategy Pattern
- D) Mediator Pattern
- **Answer:** C) Strategy Pattern

77. **In a gaming application, which pattern would be best for implementing player commands like move, attack, and defend?**

- A) Observer Pattern
- B) Command Pattern
- C) Strategy Pattern
- D) Mediator Pattern
- **Answer:** B) Command Pattern

78. **Which pattern would be suitable for managing communication between multiple subsystems in a large application?**

- A) Observer Pattern
- B) Mediator Pattern
- C) Command Pattern
- D) Chain of Responsibility Pattern
- **Answer:** B) Mediator Pattern

79. **Which pattern would be most effective for implementing a logging system that can handle different log levels (info, warning, error)?**

- A) Observer Pattern
- B) Strategy Pattern
- C) Chain of Responsibility Pattern
- D) Command Pattern
- **Answer:** C) Chain of Responsibility Pattern

80. **Which pattern would you use to decouple the request sender from the request processing logic in a remote control system?**

- A) Strategy Pattern
- B) Command Pattern
- C) Observer Pattern
- D) Mediator Pattern
- **Answer:** B) Command Pattern

Conceptual Behavioral Pattern Questions

81. **Which pattern involves encapsulating requests as objects?**
- A) Command Pattern
- B) Observer Pattern
- C) Strategy Pattern
- D) Mediator Pattern
- **Answer:** A) Command Pattern

82. **In which pattern does a subject maintain a list of dependents and notify them of changes?**
- A) Strategy Pattern
- B) Command Pattern
- C) Observer Pattern
- D) Chain of Responsibility Pattern
- **Answer:** C) Observer Pattern

83. **Which pattern focuses on allowing multiple handlers to process a request?**
- A) Chain of Responsibility Pattern
- B) Command Pattern
- C) Mediator Pattern
- D) Observer Pattern
- **Answer:** A) Chain of Responsibility Pattern

84. **Which pattern eliminates direct communication between objects by introducing a central coordinating object?**

- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Mediator Pattern
- D) Command Pattern
- **Answer:** C) Mediator Pattern

85. **Which pattern is most likely to be used when there are multiple potential algorithms for performing a task?**

- A) Chain of Responsibility Pattern
- B) Strategy Pattern
- C) Observer Pattern
- D) Command Pattern
- **Answer:** B) Strategy Pattern

86. **Which pattern would you choose if you need to undo a series of user actions?**

- A) Observer Pattern
- B) Strategy Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** C) Command Pattern

87. **Which pattern is best suited for scenarios where an object's state needs to be observed by multiple other objects?**

- A) Command Pattern
- B) Chain of Responsibility Pattern
- C) Observer Pattern
- D) Mediator Pattern
- **Answer:** C) Observer Pattern

88. **In which pattern can you dynamically add or remove handlers to process requests?**

- A) Command Pattern
- B) Chain of Responsibility Pattern
- C) Observer Pattern
- D) Strategy Pattern
- **Answer:** B) Chain of Responsibility Pattern

89. **Which pattern is ideal for reducing the number of communication channels between objects?**

- A) Strategy Pattern
- B) Observer Pattern
- C) Mediator Pattern
- D) Command Pattern
- **Answer:** C) Mediator Pattern

90. **Which pattern could you use to handle events in a GUI application, such as button clicks?**

- A) Chain of Responsibility Pattern
- B) Command Pattern
- C) Strategy Pattern
- D) Observer Pattern
- **Answer:** B) Command Pattern

Behavioral Pattern Extensions

91. **Which pattern might be used in an application where user inputs must be validated by multiple independent components?**

- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Strategy Pattern
- D) Mediator Pattern
- **Answer:** B) Chain of Responsibility Pattern

92. **Which pattern could be used to simplify the communication between a set of objects in a complex system?**

- A) Mediator Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Chain of Responsibility Pattern
- **Answer:** A) Mediator Pattern

93. **Which pattern allows a client to issue requests without knowing the specifics of how those requests will be handled?**

- A) Chain of Responsibility Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** C) Command Pattern

94. **Which pattern is best for implementing a feature where multiple objects need to react to changes in another object's state?**

- A) Strategy Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Chain of Responsibility Pattern
- **Answer:** B) Observer Pattern

95. **Which pattern should you use if you want to select different algorithms for a specific task at runtime?**

- A) Observer Pattern
- B) Command Pattern
- C) Strategy Pattern
- D) Mediator Pattern
- **Answer:** C) Strategy Pattern

96. **Which pattern would you implement if you need to execute operations in a specific sequence where each operation depends on the previous one?**

- A) Observer Pattern
- B) Chain of Responsibility Pattern
- C) Command Pattern
- D) Mediator Pattern
- **Answer:** B) Chain of Responsibility Pattern

97. **Which pattern could help you manage complex interactions between multiple objects while avoiding tight coupling?**

- A) Strategy Pattern
- B) Observer Pattern
- C) Mediator Pattern
- D) Command Pattern
- **Answer:** C) Mediator Pattern

98. **Which pattern would you choose if you needed a way to switch between different behaviors at runtime without altering the client code?**

- A) Strategy Pattern
- B) Observer Pattern
- C) Chain of Responsibility Pattern
- D) Command Pattern
- **Answer:** A) Strategy Pattern

99. **Which pattern would be most suitable for implementing a mechanism where user actions are queued and processed one after the other?**

- A) Observer Pattern
- B) Command Pattern
- C) Strategy Pattern
- D) Chain of Responsibility Pattern
- **Answer:** B) Command Pattern

100. **Which pattern would help you centralize control logic and reduce the number of connections between interacting objects in a large system?**

- A) Mediator Pattern
- B) Observer Pattern
- C) Command Pattern
- D) Strategy Pattern
- **Answer:** A) Mediator Pattern