### MCQ Set on Structural Design Patterns in C#

#### 1. Which of the following patterns is primarily used to decouple an abstraction from its implementation?

- A) Adapter Pattern

- B) Bridge Pattern

- C) Composite Pattern

- D) Facade Pattern

- \*\*Answer:\*\* B) Bridge Pattern

#### 2. The Adapter Pattern is used to:

- A) Create a simplified interface to a complex subsystem

- B) Allow incompatible interfaces to work together

- C) Treat individual objects and compositions of objects uniformly

- D) Reduce the cost of creating many similar objects

- \*\*Answer:\*\* B) Allow incompatible interfaces to work together

#### 3. Which pattern involves a hierarchy of classes where the leaf and composite classes implement the same interface?

- A) Proxy Pattern

- B) Adapter Pattern

- C) Composite Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Composite Pattern

#### 4. In the Decorator Pattern, the key feature is:

- A) Dynamic addition of behavior to an object

- B) Decoupling an abstraction from its implementation

- C) Providing a simplified interface to a complex system

- D) Allowing incompatible interfaces to work together

- \*\*Answer:\*\* A) Dynamic addition of behavior to an object

#### 5. What is the primary purpose of the Flyweight Pattern?

- A) Simplify complex subsystems

- B) Share as much data as possible to reduce memory usage

- C) Allow incompatible interfaces to work together

- D) Provide a proxy to control access to an object

- \*\*Answer:\*\* B) Share as much data as possible to reduce memory usage

#### 6. The Facade Pattern is most appropriate when:

- A) You need to reduce memory usage by sharing data

- B) You want to simplify interaction with a complex subsystem

- C) You need to add responsibilities to objects dynamically

- D) You need to decouple abstraction from its implementation

- \*\*Answer:\*\* B) You want to simplify interaction with a complex subsystem

#### 7. In the Proxy Pattern, the proxy class:

- A) Provides a simplified interface to a complex system

- B) Controls access to the real object

- C) Combines multiple objects into a single entity

- D) Allows incompatible interfaces to work together

- \*\*Answer:\*\* B) Controls access to the real object

#### 8. Which pattern can be considered a structural variant of the Adapter Pattern?

- A) Decorator Pattern

- B) Bridge Pattern

- C) Facade Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* D) Proxy Pattern

#### 9. Which of the following is not a structural design pattern?

- A) Factory Method Pattern

- B) Composite Pattern

- C) Bridge Pattern

- D) Adapter Pattern

- \*\*Answer:\*\* A) Factory Method Pattern

#### 10. The main advantage of the Composite Pattern is:

- A) It allows for complex object creation processes

- B) It treats individual objects and compositions uniformly

- C) It provides a proxy for controlling access to an object

- D) It enables decoupling of abstraction from implementation

- \*\*Answer:\*\* B) It treats individual objects and compositions uniformly

#### 11. The Bridge Pattern is useful when:

- A) You need to share as much data as possible

- B) You want to dynamically add responsibilities to objects

- C) You want to separate abstraction and implementation

- D) You need to provide a simplified interface to a complex system

- \*\*Answer:\*\* C) You want to separate abstraction and implementation

#### 12. Which pattern would you use to control access to an object, for example, loading a heavy object only when it's really needed?

- A) Facade Pattern

- B) Composite Pattern

- C) Proxy Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Proxy Pattern

#### 13. Which of the following patterns would you use to treat individual objects and groups of objects in the same way?

- A) Composite Pattern

- B) Adapter Pattern

- C) Bridge Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* A) Composite Pattern

#### 14. The Facade Pattern is most similar to:

- A) Adapter Pattern

- B) Proxy Pattern

- C) Composite Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* A) Adapter Pattern

#### 15. Which pattern helps in managing a large number of fine-grained objects efficiently?

- A) Facade Pattern

- B) Composite Pattern

- C) Flyweight Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* C) Flyweight Pattern

#### 16. The Decorator Pattern allows for:

- A) Adding new functionality to existing objects without altering their structure

- B) Providing a surrogate for another object to control access

- C) Treating individual objects and groups of objects in a uniform manner

- D) Sharing as much data as possible among similar objects

- \*\*Answer:\*\* A) Adding new functionality to existing objects without altering their structure

#### 17. In which pattern do you create an interface to hide the complexity of a subsystem?

- A) Facade Pattern

- B) Adapter Pattern

- C) Composite Pattern

- D) Bridge Pattern

- \*\*Answer:\*\* A) Facade Pattern

#### 18. Which pattern is used when an object’s structure needs to be altered dynamically?

- A) Bridge Pattern

- B) Adapter Pattern

- C) Decorator Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* C) Decorator Pattern

#### 19. Which pattern is often used in GUI toolkits to enable or disable functionality at runtime?

- A) Facade Pattern

- B) Proxy Pattern

- C) Composite Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* D) Decorator Pattern

#### 20. Which pattern is used to make a complex system easier to use by creating a simplified interface?

- A) Facade Pattern

- B) Composite Pattern

- C) Adapter Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* A) Facade Pattern

---

### 21. The main purpose of the Adapter Pattern is to:

- A) Decouple an abstraction from its implementation

- B) Provide a simplified interface to a complex subsystem

- C) Allow incompatible interfaces to work together

- D) Add responsibilities to objects dynamically

- \*\*Answer:\*\* C) Allow incompatible interfaces to work together

### 22. Which of the following patterns is best suited for managing a family of related algorithms?

- A) Adapter Pattern

- B) Bridge Pattern

- C) Composite Pattern

- D) Strategy Pattern

- \*\*Answer:\*\* D) Strategy Pattern

### 23. The Flyweight Pattern helps reduce memory usage by:

- A) Reusing objects that share similar states

- B) Creating a simplified interface to complex systems

- C) Controlling access to objects

- D) Decoupling abstraction from implementation

- \*\*Answer:\*\* A) Reusing objects that share similar states

### 24. In the Bridge Pattern, the `abstraction` and `implementation` are:

- A) Completely independent and cannot interact

- B) Combined into a single class

- C) Decoupled and can evolve independently

- D) Strongly coupled and cannot be changed independently

- \*\*Answer:\*\* C) Decoupled and can evolve independently

### 25. Which pattern allows objects to be treated as "lightweight" or "heavyweight" depending on their state and context?

- A) Composite Pattern

- B) Flyweight Pattern

- C) Facade Pattern

- D) Adapter Pattern

- \*\*Answer:\*\* B) Flyweight Pattern

### 26. In the Proxy Pattern, a virtual proxy:

- A) Acts as a placeholder for another object and controls its access

- B) Creates and initializes an object only when it's needed

- C) Manages complex subsystems and simplifies their interfaces

- D) Treats groups of objects and individual objects uniformly

- \*\*Answer:\*\* B) Creates and initializes an object only when it's needed

### 27. Which of the following patterns is useful for making a hierarchy of objects where some objects are composites of other objects?

- A) Adapter Pattern

- B) Composite Pattern

- C) Proxy Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 28. The Decorator Pattern is best applied when:

- A) You need to extend functionality without modifying existing code

- B) You need to hide the complexity of a subsystem

- C) You need to share common data across multiple objects

- D) You need to treat individual objects and compositions uniformly

- \*\*Answer:\*\* A) You need to extend functionality without modifying existing code

### 29. Which pattern is most appropriate for simplifying the interface to a subsystem with many classes?

- A) Proxy Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 30. The main goal of the Composite Pattern is to:

- A) Create large numbers of small objects efficiently

- B) Allow an object to alter its behavior when its internal state changes

- C) Treat individual objects and compositions of objects uniformly

- D) Provide a surrogate for another object to control access

- \*\*Answer:\*\* C) Treat individual objects and compositions of objects uniformly

### 31. The Adapter Pattern is typically used when:

- A) You need to add responsibilities to objects dynamically

- B) You want to create an object that represents another object

- C) You need to make two incompatible interfaces work together

- D) You need to decouple an abstraction from its implementation

- \*\*Answer:\*\* C) You need to make two incompatible interfaces work together

### 32. The primary advantage of the Bridge Pattern is:

- A) It simplifies the interface to complex subsystems

- B) It allows adding functionality without modifying existing code

- C) It decouples an abstraction from its implementation

- D) It shares data among large numbers of similar objects

- \*\*Answer:\*\* C) It decouples an abstraction from its implementation

### 33. Which of the following patterns uses composition over inheritance to add functionality to objects?

- A) Decorator Pattern

- B) Adapter Pattern

- C) Facade Pattern

- D) Composite Pattern

- \*\*Answer:\*\* A) Decorator Pattern

### 34. The Facade Pattern is most useful when:

- A) You need to manage a large number of fine-grained objects

- B) You need to provide a simplified interface to a complex system

- C) You need to dynamically change an object's behavior

- D) You need to decouple an abstraction from its implementation

- \*\*Answer:\*\* B) You need to provide a simplified interface to a complex system

### 35. In the Flyweight Pattern, intrinsic and extrinsic states are:

- A) Both stored within the flyweight object

- B) Stored in different objects to minimize memory usage

- C) Combined to create a single object for efficiency

- D) Not related to each other

- \*\*Answer:\*\* B) Stored in different objects to minimize memory usage

### 36. Which pattern would you use to ensure that a complex subsystem can be accessed easily by clients?

- A) Adapter Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 37. The Proxy Pattern is most suitable when:

- A) You need to treat objects and their compositions uniformly

- B) You need to provide a surrogate or placeholder for another object

- C) You need to add functionality to objects dynamically

- D) You need to share data among large numbers of similar objects

- \*\*Answer:\*\* B) You need to provide a surrogate or placeholder for another object

### 38. The Decorator Pattern allows:

- A) A class to inherit from multiple classes

- B) The modification of an object's behavior at runtime without changing the object itself

- C) The sharing of intrinsic data among multiple objects

- D) The decoupling of an abstraction from its implementation

- \*\*Answer:\*\* B) The modification of an object's behavior at runtime without changing the object itself

### 39. Which pattern is commonly used in GUI frameworks to add additional behavior to a window or component?

- A) Facade Pattern

- B) Adapter Pattern

- C) Decorator Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* C) Decorator Pattern

### 40. The Composite Pattern is particularly useful in:

- A) Creating user interface components

- B) Sharing data among objects

- C) Adding functionality to an object dynamically

- D) Controlling access to another object

- \*\*Answer:\*\* A) Creating user interface components

### 41. Which of the following best describes the intent of the Facade Pattern?

- A) Provide a simplified interface to a complex subsystem

- B) Allow incompatible interfaces to work together

- C) Treat individual objects and compositions of objects uniformly

- D) Share as much data as possible among similar objects

- \*\*Answer:\*\* A) Provide a simplified interface to a complex subsystem

### 42. The Flyweight Pattern minimizes memory usage by:

- A) Storing shared data in a single object and extrinsic data separately

- B) Using inheritance to create specialized objects

- C) Providing a surrogate for another object

- D) Treating individual objects and compositions uniformly

- \*\*Answer:\*\* A) Storing shared data in a single object and extrinsic data separately

### 43. The Adapter Pattern is often used to:

- A) Convert the interface of a class into another interface clients expect

- B) Create a simplified interface to a complex system

- C) Control access to another object

- D) Treat individual objects and compositions uniformly

- \*\*Answer:\*\* A) Convert the interface of a class into another interface clients expect

### 44. Which pattern is useful when an object needs to be able to extend its behavior dynamically?

- A) Adapter Pattern

- B) Facade Pattern

- C) Decorator Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Decorator Pattern

### 45. The Composite Pattern is most similar to which other pattern in intent?

- A) Flyweight Pattern

- B) Proxy Pattern

- C) Adapter Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* D) Decorator Pattern

### 46. Which pattern involves creating a simplified interface to a subsystem that has many interacting components?

- A) Bridge Pattern

- B) Adapter Pattern

- C) Facade Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 47. The Bridge Pattern allows for:

- A) Hiding the complexity of a subsystem

- B) Decoupling an abstraction from its implementation so they can vary independently

- C) Controlling access to another object

- D) Sharing data among similar objects

- \*\*Answer:\*\* B) Decoupling an abstraction from its implementation so they can vary independently

### 48. Which pattern is used to provide a surrogate or placeholder for another object to control access to it?

- A) Adapter Pattern

- B) Facade Pattern

- C) Proxy Pattern

- D) Composite Pattern

- \*\*Answer:\*\* C) Proxy Pattern

### 49. Which pattern is used to treat a group of objects as a single object?

- A) Adapter Pattern

- B) Facade Pattern

- C) Composite Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* C) Composite Pattern

### 50. The main difference between the Adapter and Facade patterns is:

- A) Adapter is used to change the interface of an existing object, while Facade is used to simplify the interface to a subsystem

- B) Adapter provides a surrogate for another object, while Facade provides access control

- C) Adapter treats objects uniformly, while Facade hides complexity

- D) Adapter is structural, while Facade is behavioral

- \*\*Answer:\*\* A) Adapter is used to change the interface of an existing object, while Facade is used to simplify the interface to a subsystem

### 51. The primary purpose of the Flyweight Pattern is:

- A) To reduce the memory footprint of complex objects

- B) To allow incompatible interfaces to work together

- C) To simplify the interaction with complex subsystems

- D) To treat individual objects and groups uniformly

- \*\*Answer:\*\* A) To reduce the memory footprint of complex objects

### 52. Which pattern is useful when you need to vary an object's behavior based on its state?

- A) State Pattern

- B) Strategy Pattern

- C) Composite Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* D) Decorator Pattern

### 53. Which pattern is typically used to create a simplified interface to a complex set of subsystems?

- A) Adapter Pattern

- B) Facade Pattern

- C) Proxy Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* B) Facade Pattern

### 54. The Bridge Pattern is designed to:

- A) Provide a single class to represent a group of objects

- B) Allow two incompatible interfaces to work together

- C) Separate an object's abstraction from its implementation

- D) Control access to a resource

- \*\*Answer:\*\* C) Separate an object's abstraction from its implementation

### 55. The Flyweight Pattern involves:

- A) Creating small, reusable objects that share common data

- B) Providing a simplified interface to a complex system

- C) Treating individual objects and groups uniformly

- D) Adding new responsibilities to an object dynamically

- \*\*Answer:\*\* A) Creating small, reusable objects that share common data

### 56. The Decorator Pattern is best suited for:

- A) Simplifying a complex system

- B) Adding behavior to an object dynamically

- C) Allowing incompatible interfaces to work together

- D) Sharing data among multiple objects

- \*\*Answer:\*\* B) Adding behavior to an object dynamically

### 57. Which pattern should be used when you want to simplify the use of a large and complex system?

- A) Adapter Pattern

- B) Flyweight Pattern

- C) Facade Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 58. The Proxy Pattern is different from the Adapter Pattern because:

- A) Proxy controls access to an object, while Adapter changes the interface

- B) Proxy adds new behavior to an object, while Adapter simplifies interfaces

- C) Proxy is used for large systems, while Adapter is for small systems

- D) Proxy creates lightweight objects, while Adapter treats groups uniformly

- \*\*Answer:\*\* A) Proxy controls access to an object, while Adapter changes the interface

### 59. Which pattern would you use to ensure that a complex system is presented to users through a simple interface?

- A) Proxy Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 60. In which pattern are objects composed into tree structures to represent part-whole hierarchies?

- A) Adapter Pattern

- B) Composite Pattern

- C) Flyweight Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 61. The main advantage of using the Bridge Pattern is:

- A) It simplifies complex systems

- B) It allows for changing implementations independently of the abstraction

- C) It adds new behavior dynamically

- D) It controls access to another object

- \*\*Answer:\*\* B) It allows for changing implementations independently of the abstraction

### 62. The Flyweight Pattern is particularly useful in scenarios where:

- A) There are large numbers of similar objects that share common data

- B) You need to control access to an object

- C) You need to dynamically change an object's behavior

- D) You need to provide a simplified interface to a complex system

- \*\*Answer:\*\* A) There are large numbers of similar objects that share common data

### 63. The Decorator Pattern is often used to:

- A) Add or modify the responsibilities of objects at runtime

- B) Simplify complex subsystems

- C) Share data among objects efficiently

- D) Control access to resources

- \*\*Answer:\*\* A) Add or modify the responsibilities of objects at runtime

### 64. Which pattern should be used when you need to allow an object to behave differently depending on its internal state?

- A) Flyweight Pattern

- B) State Pattern

- C) Decorator Pattern

- D) Composite Pattern

- \*\*Answer:\*\* C) Decorator Pattern

### 65. In the Facade Pattern, the facade:

- A) Provides a simplified interface to a complex system

- B) Treats groups of objects and individual objects uniformly

- C) Controls access to an object

- D) Decouples an abstraction from its implementation

- \*\*Answer:\*\* A) Provides a simplified interface to a complex system

### 66. The Adapter Pattern is different from the Facade Pattern because:

- A) Adapter changes the interface of an existing object, while Facade simplifies an interface to a subsystem

- B) Adapter treats objects uniformly, while Facade hides complexity

- C) Adapter adds behavior dynamically, while Facade decouples abstractions

- D) Adapter is for complex systems, while Facade is for simple ones

- \*\*Answer:\*\* A) Adapter changes the interface of an existing object, while Facade simplifies an interface to a subsystem

### 67. Which pattern is best suited for reducing the number of objects created in a system by sharing data?

- A) Flyweight Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* A) Flyweight Pattern

### 68. Which pattern allows for treating individual objects and compositions of objects uniformly?

- A) Adapter Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 69. The Proxy Pattern can be used to:

- A) Control access to a resource-intensive object

- B) Simplify the interface to a complex system

- C) Add new functionality to an object dynamically

- D) Share data among multiple objects

- \*\*Answer:\*\* A) Control access to a resource-intensive object

### 70. The main benefit of the Composite Pattern is:

- A) It allows treating individual objects and groups of objects the same way

- B) It reduces memory usage by sharing data

- C) It adds new behavior to objects dynamically

- D) It decouples an abstraction from its implementation

- \*\*Answer:\*\* A) It allows treating individual objects and groups of objects the same way

### 71. Which pattern is designed to handle cases where multiple related objects need to be treated as a single object?

- A) Proxy Pattern

- B) Composite Pattern

- C) Facade Pattern

- D) Adapter Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 72. The Facade Pattern is used to:

- A) Simplify the interface to a complex system

- B) Share data among multiple objects

- C) Add behavior to objects dynamically

- D) Control access to an object

- \*\*Answer:\*\* A) Simplify the interface to a complex system

### 73. The main purpose of the Flyweight Pattern is to:

- A) Minimize memory usage by sharing as much data as possible with other similar objects

- B) Simplify complex systems

- C) Allow objects to behave differently based on their state

- D) Add new functionality to objects at runtime

- \*\*Answer:\*\* A) Minimize memory usage by sharing as much data as possible with other similar objects

### 74. Which pattern provides a way to attach additional responsibilities to an object at runtime?

- A) Composite Pattern

- B) Facade Pattern

- C) Flyweight Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* D) Decorator Pattern

### 75. In the Composite Pattern, a leaf is:

- A) An individual object in a composition

- B) A group of objects treated as a single entity

- C) An interface for simplified interaction

- D) A proxy object that controls access

- \*\*Answer:\*\* A) An individual object in a composition

### 76. The Decorator Pattern allows:

- A) Adding responsibilities to an object dynamically

- B) Creating lightweight objects for efficient memory usage

- C) Simplifying a complex subsystem

- D) Treating groups of objects uniformly

- \*\*Answer:\*\* A) Adding responsibilities to an object dynamically

### 77. Which pattern is most appropriate when you need to ensure that a complex system can be accessed easily by clients?

- A) Proxy Pattern

- B) Facade Pattern

- C) Composite Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* B) Facade Pattern

### 78. The primary difference between the Adapter and Decorator patterns is:

- A) Adapter changes the interface, while Decorator adds functionality

- B) Adapter simplifies interaction, while Decorator changes behavior

- C) Adapter decouples implementation, while Decorator treats groups uniformly

- D) Adapter manages resources, while Decorator simplifies systems

- \*\*Answer:\*\* A) Adapter changes the interface, while Decorator adds functionality

### 79. The Flyweight Pattern is particularly effective when:

- A) A large number of objects share a common intrinsic state

- B) You need to add new behavior to an object

- C) You need to control access to an object

- D) You need to simplify a complex subsystem

- \*\*Answer:\*\* A) A large number of objects share a common intrinsic state

### 80. Which pattern allows you to treat objects and their compositions in a uniform manner?

- A) Facade Pattern

- B) Composite Pattern

- C) Flyweight Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 81. The Proxy Pattern is most suitable when:

- A) You need to provide a surrogate or placeholder for another object

- B) You need to add new responsibilities to an object dynamically

- C) You need to share as much data as possible among similar objects

- D) You need to simplify the interface to a complex system

- \*\*Answer:\*\* A) You need to provide a surrogate or placeholder for another object

### 82. Which pattern is best suited for dynamically adding new functionality to objects without affecting other objects?

- A) Flyweight Pattern

- B) Composite Pattern

- C) Decorator Pattern

- D) Facade Pattern

- \*\*Answer:\*\* C) Decorator Pattern

### 83. The primary intent of the Composite Pattern is to:

- A) Allow clients to treat individual objects and compositions of objects uniformly

- B) Minimize memory usage by sharing data

- C) Provide a simplified interface to a complex system

- D) Control access to a resource-intensive object

- \*\*Answer:\*\* A) Allow clients to treat individual objects and compositions of objects uniformly

### 84. Which pattern provides a way to create a simplified interface to a complex system with many interacting components?

- A) Adapter Pattern

- B) Facade Pattern

- C) Proxy Pattern

- D) Composite Pattern

- \*\*Answer:\*\* B) Facade Pattern

### 85. The Decorator Pattern is different from inheritance because:

- A) It allows behavior to be added to individual objects dynamically

- B) It simplifies the interaction with complex subsystems

- C) It treats objects and compositions uniformly

- D) It controls access to objects

- \*\*Answer:\*\* A) It allows behavior to be added to individual objects dynamically

### 86. The Bridge Pattern is most useful when:

- A) You want to separate the abstraction from its implementation

- B) You need to control access to a resource

- C) You need to share data among objects

- D) You need to simplify a complex system's interface

- \*\*Answer:\*\* A) You want to separate the abstraction from its implementation

### 87. Which pattern would you use when you have a complex subsystem and want to provide a simpler interface to it?

- A) Adapter Pattern

- B) Proxy Pattern

- C) Flyweight Pattern

- D) Facade Pattern

- \*\*Answer:\*\* D) Facade Pattern

### 88. The Flyweight Pattern is most effective when:

- A) There are many similar objects that can share common data

- B) You need to provide a simplified interface to a subsystem

- C) You need to treat objects and compositions uniformly

- D) You need to dynamically add functionality to objects

- \*\*Answer:\*\* A) There are many similar objects that can share common data

### 89. In the Proxy Pattern, the proxy class:

- A) Controls access to another object

- B) Simplifies the interface to a complex system

- C) Treats individual objects and compositions uniformly

- D) Adds new behavior to an object dynamically

- \*\*Answer:\*\* A) Controls access to another object

### 90. The main advantage of using the Composite Pattern is:

- A) It simplifies complex systems

- B) It allows for uniform treatment of individual and composite objects

- C) It adds new responsibilities to an object dynamically

- D) It reduces memory usage by sharing data

- \*\*Answer:\*\* B) It allows for uniform treatment of individual and composite objects

### 91. Which pattern provides a way to add new responsibilities to an object at runtime?

- A) Adapter Pattern

- B) Decorator Pattern

- C) Facade Pattern

- D) Flyweight Pattern

- \*\*Answer:\*\* B) Decorator Pattern

### 92. The Flyweight Pattern is used to:

- A) Minimize memory usage by sharing as much data as possible

- B) Simplify the interface to a complex system

- C) Treat individual objects and compositions uniformly

- D) Add new responsibilities to an object dynamically

- \*\*Answer:\*\* A) Minimize memory usage by sharing as much data as possible

### 93. Which pattern is useful when you need to provide a simple interface to a subsystem that has many interfaces?

- A) Composite Pattern

- B) Adapter Pattern

- C) Facade Pattern

- D) Proxy Pattern

- \*\*Answer:\*\* C) Facade Pattern

### 94. The primary benefit of the Decorator Pattern is:

- A) It allows new behavior to be added to an object dynamically

- B) It simplifies complex systems

- C) It controls access to another object

- D) It reduces memory usage

- \*\*Answer:\*\* A) It allows new behavior to be added to an object dynamically

### 95. The Bridge Pattern is best applied when:

- A) You need to separate the abstraction of a class from its implementation

- B) You need to treat individual objects and compositions uniformly

- C) You need to add new responsibilities to an object dynamically

- D) You need to minimize memory usage by sharing data

- \*\*Answer:\*\* A) You need to separate the abstraction of a class from its implementation

### 96. The Proxy Pattern is often used in:

- A) Virtual proxies to delay the creation of expensive objects

- B) Simplifying complex subsystem interactions

- C) Treating groups of objects uniformly

- D) Adding new behavior to objects at runtime

- \*\*Answer:\*\* A) Virtual proxies to delay the creation of expensive objects

### 97. Which pattern would you use to treat individual objects and compositions of objects uniformly in a hierarchical structure?

- A) Facade Pattern

- B) Composite Pattern

- C) Flyweight Pattern

- D) Adapter Pattern

- \*\*Answer:\*\* B) Composite Pattern

### 98. The primary difference between the Flyweight and Composite patterns is:

- A) Flyweight focuses on minimizing memory usage, while Composite focuses on hierarchical structures

- B) Flyweight simplifies complex systems, while Composite controls access

- C) Flyweight adds new behavior, while Composite decouples implementation

- D) Flyweight controls access, while Composite treats objects uniformly

- \*\*Answer:\*\* A) Flyweight focuses on minimizing memory usage, while Composite focuses on hierarchical structures

### 99. The Facade Pattern is useful for:

- A) Simplifying the interface to a complex system

- B) Adding new behavior to objects dynamically

- C) Controlling access to a resource-intensive object

- D) Treating individual objects and compositions uniformly

- \*\*Answer:\*\* A) Simplifying the interface to a complex system

### 100. Which pattern allows you to separate an abstraction from its implementation so that the two can evolve independently?

- A) Flyweight Pattern

- B) Proxy Pattern

- C) Bridge Pattern

- D) Decorator Pattern

- \*\*Answer:\*\* C) Bridge Pattern