

26 Lecture - CS504

Important Mcqs

Q: What are design patterns in software development? a) Detailed implementation guidelines for specific programming languages. b) Reusable solutions to common software design problems. c) Comprehensive project management methodologies. d) Documentation templates for software projects. **Solution: b) Reusable solutions to common software design problems.**

Q: Which of the following is a benefit of using design patterns? a) Increased software complexity. b) Reduced code flexibility. c) Improved code maintainability. d) Limited code reusability. **Solution: c) Improved code maintainability.**

Q: Design patterns are intended to be: a) Specific to individual programming languages. b) Adapted to each software project independently. c) Reusable across various software projects. d) Only used in small-scale applications. **Solution: c) Reusable across various software projects.**

Q: The Singleton pattern ensures that a class has: a) Multiple instances with shared data. b) Multiple instances with separate data. c) Only one instance throughout the application. d) Multiple instances with limited access. **Solution: c) Only one instance throughout the application.**

Q: Which design pattern promotes loose coupling between objects? a) Adapter pattern. b) Observer pattern. c) Facade pattern. d) Singleton pattern. **Solution: b) Observer pattern.**

Q: In the Factory Method pattern, the responsibility of object creation is: a) Assigned to the client code. b) Delegated to a factory class. c) Shared between multiple classes. d) Ignored, as objects are created implicitly. **Solution: b) Delegated to a factory class.**

Q: The Strategy pattern enables: a) Objects to communicate with each other. b) Dynamically changing algorithms or behaviors. c) Objects to share their data with others. d) Objects to hide their internal structure. **Solution: b) Dynamically changing algorithms or behaviors.**

Q: Which design pattern provides a simple interface to a complex subsystem? a) Singleton pattern. b) Facade pattern. c) Factory Method pattern. d) Adapter pattern. **Solution: b) Facade pattern.**

Q: The Chain of Responsibility pattern is used to: a) Establish relationships between classes in an inheritance hierarchy. b) Create a chain of interconnected objects to process a request. c) Ensure that only one instance of a class exists throughout the application. d) Simplify the creation of objects in complex systems. **Solution: b) Create a chain of interconnected objects to process a request.**

Q: The Decorator pattern allows: a) Adding new functionalities to objects dynamically. b) Sharing data between multiple objects. c) Preventing multiple instances of a class. d) Restricting access to certain classes. **Solution: a) Adding new functionalities to objects dynamically.**