Interview Questions on SOLID Design Principles

1. Can you name the five SOLID principles?
2. Can you explain briefly the SOLID design principles?
3. How do you ensure Single Responsibility Principle (SRP) in your code?
4. How do you implement Open-Closed Principle (OCP) in your code?
5. How do you ensure Liskov Substitution Principle (LSP) in your code?
6. How do you apply Interface Segregation Principle (ISP) in your code?
7. How do you apply Dependency Inversion Principle (DIP) in your code?
8. How do SOLID principles help in achieving maintainable code?
9. What are the common violations of SOLID principles?
10. What are the advantages of following SOLID principles in software development?
11. How Design Principles are different from Design Patterns?
12. What do you understand about coupling and cohesion in software development?
13. How do you reduce coupling and increase cohesion in your code?
14. What are some best practices for avoiding duplicate code?
15. Why is tight coupling bad?
16. What are some ways to avoid using multiple inheritance in programming?
17. Do you prefer loose or tight coupling in your programs? Why?
18. What is the difference between object oriented programming and procedural programming?
19. What are some cases where high cohesion can be seen as negative?
20. What is OCP (Open/Closed Principle) and why is it important?
21. What are the benefits of loose coupling in software development?
22. How do you deal with tight coupling in your code?
23. What are some disadvantages of tightly coupled classes?
24. How would you identify if a codebase violates the Single Responsibility Principle (SRP)?
25. How would you refactor a codebase that violates the Open-Closed Principle (OCP)?
26. How would you ensure adherence to the Liskov Substitution Principle (LSP) when designing a new class hierarchy?
27. How would you refactor a codebase to adhere to the Interface Segregation Principle (ISP)?
28. How would you design a dependency injection container to adhere to the Dependency Inversion Principle (DIP)?
29. How do you apply SOLID principles to achieve scalability in a distributed system?

30. How would you design a system that adheres to SOLID principles and allows for easy experimentation and prototyping?
31. How would you identify if a codebase violates SOLID principles due to its design patterns?
32. How do you balance SOLID principles with the need for performance optimization in a high-performance application?
33. How do you ensure adherence to SOLID principles when integrating third-party libraries into your codebase?