C# OOP Interview Questions

Basic Concepts 1. What is Object-Oriented Programming (OOP)? 2. What are the four pillars of OOP? 3. What is a class and what is an object? 4. What is the difference between a class and an object? 5. What is encapsulation and why is it important? 6. What is abstraction and how is it implemented in C#? 7. What is inheritance and what are its types in C#? 8. What is polymorphism and what are its types? 9. What is the difference between compile-time and run-time polymorphism? 10. What is the difference between overloading and overriding?

Advanced Concepts 11. What is an abstract class and when should it be used? 12. What is an interface and when should it be used? 13. What is the difference between an abstract class and an interface? 14. What is a sealed class and when should it be used? 15. What is a static class and static member? 16. What is the difference between const and readonly? 17. What is a constructor and a destructor? 18. What is the difference between a default constructor and a parameterized constructor? 19. What is a copy constructor and does C# support it? 20. What is the difference between this and base keywords? 21. What is a partial class and when would you use it? 22. What is a nested class in C#? 23. What is the difference between value types and reference types? 24. What is method hiding and how does it differ from method overriding? 25. What are properties in C# and how do they relate to encapsulation? 26. What is an indexer and how is it used? 27. What is operator overloading and how is it implemented? 28. What is the difference between shallow copy and deep copy? 29. What is the difference between virtual, override, and new keywords in C#? 30. What is the difference between an event and a delegate? 31. What is a delegate and how is it used? 32. What are lambda expressions and how are they related to delegates? 33. What is the difference between an interface and a delegate? 34. What is multiple inheritance and how is it handled in C#? 35. What is the difference between class inheritance and interface implementation? 36. What is object composition and how does it differ from inheritance? 37. What is dependency injection and how does it relate to OOP? 38. What is the difference between aggregation and composition? 39. What is the difference between static and instance members? 40. How does garbage collection work in C#?