1. **What is Object-Oriented Programming (OOP)?**

**Answer:** Object-Oriented Programming is the methodology to write a code in way that a code should have Maintainable, Feasible and Reusable easily. Object-Oriented Programming include feature like..

* Classes
* Creating Object
* Inheritance
* Polymorphism
* Interface
* Abstraction

**2. Explain the four main principles of OOP.**

**3. What is the difference between a class and an object?**

**4. Define the concept of inheritance in OOP. Provide an example.**

**5. What is polymorphism? How does it help in OOP?**

**6. Explain the difference between method overloading and method overriding.**

**7. What is encapsulation, and why is it important in OOP?**

**8. How does abstraction differ from encapsulation in OOP?**

**9. What is a constructor in OOP, and what is its purpose?**

**10. What is a destructor, and when is it used in OOP?**

**11. What are access modifiers in OOP? List some common access modifiers.**

**12. What is the difference between public, private, and protected access modifiers?**

**13. How does multiple inheritance work in OOP?**

**14. What is a base class and a derived class in OOP?**

**15. What is the difference between static and instance methods in OOP?**

**16. What is the role of the this keyword in OOP?**

**17. What is an interface in OOP, and how is it different from an abstract class?**

**18. What are abstract methods, and how are they used in OOP?**

**19. Explain the concept of dynamic method dispatch in OOP.**

**20. What is the difference between composition and aggregation in OOP?**

**21. How do you achieve runtime polymorphism in OOP?**

**22. What is the purpose of getters and setters in OOP?**

**23. How does object composition differ from object inheritance?**

**24. What is the significance of the final keyword in OOP?**

**25. What is a singleton class, and when would you use it?**

**26. Explain the concept of method chaining in OOP.**

**27. What is the difference between shallow copy and deep copy of objects?**

**28. How does the garbage collection mechanism work in OOP?**

**29. What is a virtual function, and how is it used in OOP?**

**30. What is the difference between static and dynamic binding?**

**31. Explain the concept of a class hierarchy in OOP.**

**32. How does inheritance support code reusability in OOP?**

**33. What are the benefits and drawbacks of multiple inheritance?**

**34. What is the purpose of the super keyword in OOP?**

**35. What is the role of polymorphism in achieving loose coupling in OOP?**

**36. What are the advantages of using interfaces in OOP?**

**37. How can you prevent inheritance in OOP?**

**38. What is the difference between a class and a struct in OOP?**

**39. What is an object reference in OOP?**

**40. What is the purpose of the toString() method in OOP?**

**41. How do you implement operator overloading in OOP?**

**42. What is a concrete class, and how is it different from an abstract class?**

**43. What is the principle of "programming to an interface" in OOP?**

**44. Explain the concept of cohesion and coupling in OOP.**

**45. What is a design pattern in OOP? Provide an example.**

**46. What is the Factory Method design pattern in OOP?**

**47. What is the Observer design pattern in OOP?**

**48. How does the Strategy design pattern work in OOP?**

**49. What is the Adapter design pattern in OOP?**

**50. What is the purpose of the Dependency Injection pattern in OOP?**