Java OOP Practical Questions

1. **Write a simple class `Car` with fields `brand` and `speed`. Create an object and assign values.**
2. **Write a constructor for a class `Book` that initializes `title` and `author`.**
3. **Demonstrate constructor overloading using a class `Student`.**
4. **Write a program that uses the `this` keyword to refer to current object fields.**
5. **Create a class with a method `display()` and call it from the main method.**
6. **Create a class `Rectangle` with a method to calculate area. Create an object and display area.**
7. **Write a program to show default values of instance variables in Java.**
8. **Create a class `Person` with private fields and public getter/setter methods (encapsulation).**
9. **Write a method that returns a value from a class. Call it and print the result.**
10. **Explain with code: What happens if no constructor is defined in a class?**

11. **Create a class `Animal` with method `sound()`. Inherit it in `Dog` and override `sound()`.**
12. **Demonstrate single inheritance with `Vehicle` -> `Car` structure.**
13. **Write code that shows constructor chaining using `super()`.**
14. **Show how the `super` keyword can be used to access parent class methods.**
15. **Create a class `A` with method `display()`. Create class `B` that extends `A` and calls `display()`.**
16. **Write a Java program to illustrate multilevel inheritance.**
17. **Create a method with the same name in parent and child class. Call using parent reference.**
18. **Write a program to demonstrate method overriding.**
19. **Use inheritance to create a class `Shape` with subclasses `Circle` and `Square`. Add area methods.**
20. **Explain: What is the role of access modifiers in inheritance? (Use code for demo)**

21. **Write a program to show method overloading in a class `Calculator`.**
22. **Write a class with an overridden method and demonstrate runtime polymorphism.**
23. **Create a parent class reference that holds child class object and calls overridden method.**
24. **Write a code to show compile-time polymorphism with `add(int, int)` and `add(double, double)`.**
25. **Explain with code: What is the difference between overloading and overriding?**
26. **Write code to demonstrate dynamic method dispatch.**
27. **Create a method `printDetails()` in both base and derived class and call it via reference.**

28. **Create an abstract class `Bank` with abstract method `getInterestRate()`. Extend it in `SBI`.**
29. **Write a program with an abstract class and implement the method in the subclass.**
30. **Create an interface `Flyable` with method `fly()`. Implement it in `Bird` class.**
31. **Demonstrate multiple interfaces using `Printable` and `Scannable`.**

32. **Write code to show interface with constants and abstract methods.**
33. **Can you instantiate an abstract class? Explain with an example.**
34. **Can a class implement multiple interfaces and extend a class? Show with syntax.**
35. **Write a program that uses an interface reference to call implemented class method.**

36. **Write a class with fields having `private`, `public`, and `protected` access. Try accessing them.**
37. **Use the `final` keyword to prevent method overriding. Show with a code example.**
38. **Create a static method and static variable. Access them without creating an object.**
39. **Write a program to count objects using a static variable.**
40. **Use the `instanceof` keyword to check object type.**
41. **Demonstrate the use of `final` class. Can it be extended? Show with explanation.**