B26. What happens when a class implements two interfaces and both declare a field (variable) with the same name?

If a type implements two interfaces, and each interface defines a method that has identical signature, then in effect there is only one method, and they are not distinguishable. If, say, the two methods have conflicting return types, then it will be a compilation error.

B27. Can a subclass instance method override a superclass static method?

No, we cannot override static methods in Java. We can declare static methods with the same signature in subclass but it is not considered overriding as there would not be any run-time polymorphism. Overriding is a feature of OOP languages like Java that is related to run-time polymorphism.

B28. Can a subclass static method hide a superclass instance method?

Static Methods If a subclass defines a static method with the same signature as a static method in the superclass, then the method in the subclass hides the one in the superclass. The distinction between hiding a static method and overriding an instance method has important implications:

B29. Can a superclass access subclass member?

No, a superclass has no knowledge of its subclasses. Yes, a subclass has access to all non private members of its superclass.

B30. Difference between object oriented and object based language.

Object-oriented languages follow all the concepts of OOPs whereas the object-based language doesn’t follow all the concepts of OOPs like inheritance and polymorphism. Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects, for example, JavaScript has window objects.

B31. Explain Diamond problem.

The Diamond Problem is an ambiguity that arises in multiple inheritance when two parent classes inherit from the same grandparent class, and both parent classes are inherited by a single child class. Without using virtual inheritance, the child class would inherit the properties of the grandparent class twice, leading to ambiguity.

B32. Why does Dart not support operator overloading?

Dart did not support overloading originally because it was a much more dynamic language where the declared types did not have any semantic effect. That made it impossible to use static type based overload resolution.

B33. What is Encapsulation in Dart?

Encapsulation In Dart Encapsulation is the mechanism that confirms critical and sensitive data which is hidden from users. To achieve encapsulation, you can make fields private and use the public getter and setter method to access and set the value of that field.

B34. Which of the Dart OOPS features promotes access protection or data hiding?

Class , object, Data Encapsulation , Inheritance , Polymorphism , abstraction.