**21. What is inheritance?**

Inheritance is a core concept of Object-Oriented Programming (OOP) where one class (child/subclass) inherits properties and methods from another class (parent/superclass). It allows code reuse and creates a hierarchical relationship between classes.

**22. Which inheritance is not supported by Dart? Why?**

Dart does not support **multiple inheritance**, meaning a class cannot extend more than one class. This is because multiple inheritance can cause ambiguity (e.g., if two parent classes have the same method name). Dart resolves this by using **mixins** for shared behavior between classes.

**B3. What is the advantage of inheritance?** The advantages of inheritance are:

* **Code reuse**: Subclasses can use the properties and methods of a superclass without redefining them.
* **Extensibility**: Existing code can be extended with new features without modifying the original code.
* **Hierarchy**: Helps in defining a natural hierarchical structure of objects.

**23. Difference between inheritance and encapsulation.**

* **Inheritance**: Focuses on code reuse by enabling a child class to inherit properties and methods from a parent class.
* **Encapsulation**: Involves wrapping data (variables) and methods (functions) into a single unit (class) and restricting access to them using access modifiers (private, public).

**B5. Difference between inheritance and abstraction.**

* **Inheritance**: Allows one class to inherit the features (methods and properties) of another class.
* **Abstraction**: Hides the implementation details and exposes only the functionality to the user, typically through abstract classes or interfaces.

**24. Difference between inheritance and polymorphism.**

* **Inheritance**: Establishes a parent-child relationship between classes, where the child class can inherit properties and methods from the parent class.
* **Polymorphism**: Allows one interface to be used for different types. For example, a method can perform different tasks based on the object that calls it.

**25. Can we override static methods in Dart?**

No, static methods in Dart cannot be overridden because they belong to the class itself, not to any instance of the class.

**26. Can we overload static methods in Dart?**

No, Dart does not support method overloading (multiple methods with the same name but different signatures) including static methods.

**27. Can a class implement more than one interface?**

Yes, in Dart, a class can implement multiple interfaces. This is done by using the implements keyword and separating interfaces with commas.

**B10. Can a class extend more than one class in Dart?**

No, Dart does not support multiple inheritance. A class can only extend one class at a time.

**28. Can an interface extend more than one interface in Dart?**

Yes, an interface (abstract class) can extend multiple interfaces using the extends keyword.

**29. What will happen if a class implements two interfaces and they both have a method with the same name and signature?**

If a class implements two interfaces that have methods with the same name and signature, the class must provide a single implementation of the method. This resolves the ambiguity.

**30. Can we pass an object of a subclass to a method expecting an object of the superclass?**

Yes, this is allowed because of the **is-a** relationship in inheritance. A subclass is considered an instance of its superclass.

**B14. Are static members inherited by subclasses?**

Static members are not inherited by subclasses. They belong to the class itself and can be accessed using the class name.

**31. What happens if the parent and the child class have a field with the same identifier?** If both the parent and child class have a field with the same name, the child class field hides the parent class field. To access the parent class field, you can use super.

**B16. Are constructors and initializers also inherited by subclasses?**

Constructors are not inherited by subclasses. A subclass must define its own constructors, although it can call a superclass's constructor using super.

**32. How do you restrict a member of a class from being inherited by its subclasses?**

In Dart, you can restrict inheritance by marking a class or method as final or using the \_ prefix (private to the library).

**33. How do you implement multiple inheritance in Dart?**

Dart doesn't support multiple inheritance directly, but you can achieve similar functionality using **mixins**, where a class can include multiple mixins.

**34. Can a class extend itself in Dart?**

No, a class cannot extend itself. This would create an infinite inheritance loop.

**35. How do you override a private method in Dart?**

Private methods (those with \_ prefix) cannot be overridden in other classes, including subclasses, as they are restricted to the library they are defined in.

**36. When to overload a method in Dart and when to override it?**

* **Overriding** is done when a subclass needs to provide a specific implementation of a method that exists in its superclass.
* **Overloading** is not supported in Dart. If you need different versions of a method, use different method names or optional parameters.

**37. What is the order of extends and implements keywords in Dart class declaration?** The extends keyword comes first, followed by the implements keyword

**38. How do you prevent overriding a Dart method without using the final modifier?**

You can prevent overriding by making the method private (prefixing with \_), which restricts it to the current library.

**39. What are the rules of method overriding in Dart?**

* The method must have the same name and parameter types as the method in the superclass.
* The return type must be the same or a subtype.
* The method must use the @override annotation.

**40. Difference between method overriding and overloading in Dart.**

* **Overriding**: Redefining a method from a superclass in a subclass to provide specific behavior.
* **Overloading**: Having multiple methods with the same name but different parameter types (not supported in Dart).

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

If two interfaces declare a field with the same name, the implementing class must provide its own definition. Dart does not allow multiple field inheritance conflicts.

**42. Can a subclass instance method override a superclass static method?**

No, instance methods cannot override static methods, as static methods belong to the class itself, not its instances.

**43. Can a subclass static method hide a superclass instance method?**

Yes, a subclass static method can hide a superclass instance method, but this practice is generally discouraged as it can lead to confusion.

**44. Can a superclass access subclass members?**

No, a superclass cannot access subclass-specific members directly. The subclass may, however, expose them through methods.

**45. Difference between object-oriented and object-based languages.**

* **Object-oriented languages**: Support all the OOP principles (like inheritance, polymorphism, abstraction, encapsulation).
* **Object-based languages**: Use objects but may not support inheritance or other OOP features.