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

* In Dart, a class can implement multiple interfaces, but it can only extend one class.

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

* Yes, in Dart, an interface can extend multiple interfaces.

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

* In Dart, if a class implements two interfaces, and both interfaces have a method with the same name and signature, it is perfectly valid.

**30. Can we pass an object of a subclass to a method expecting an object of the super class? Are static members inherited to sub classes?**

* Yes, in object-oriented programming languages like Dart, you can pass an object of a subclass to a method that expects an object of the superclass.
* No, static members are not inherited by subclasses in Dart.

**31. What happens if the parent and the child class have a field with same identifier? Are constructors and initializers also inherited to sub classes?**

* In Dart, if both the parent and the child class have a field with the same identifier, the child class's field will override the parent class's field.
* In Dart, constructors are not inherited by subclasses.
* Also In Dart, initializer expressions are not inherited by subclasses.

**32. How do you restrict a member of a class from inheriting by its sub classes?**

* Can use access specifier private and Make the Member Private.

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

* While Dart does not support multiple class inheritance, you can use abstraction to define a contract that classes must follow them.

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

* No, in Dart, a class cannot extend itself.

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

* In Dart, it's not possible to directly override a private method in the traditional sense because private methods are not visible outside the class where they are defined. However, you can achieve a form of behaviour modification by defining a method in the subclass with the same name, effectively overriding the private method from the superclass.

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

In Dart:

* **Method Overloading:** Dart doesn't have traditional method overloading. You can use optional or named parameters for similar flexibility.
* **Method Overriding:** Override a method in a subclass when you want to provide a specific implementation for a method declared in its superclass. Use method overriding for customization and extension in the context of class inheritance.

Top of Form

**37. What the order is of extends and implements keyword on Dart class declaration?**

In Dart, the order is:

* **extends** comes before **implements** in a class declaration.

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

* To prevent method overriding in Dart without using the final modifier, use the @override annotation in the superclass method. This signals that the method is not meant to be overridden, and attempting to override it in a subclass will result in a compile-time error.

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

In Dart, the rules for method overriding are:

1. Method Signature:
2. Use of @override:
3. Inheritance:
4. Visibility:
5. Covariant Parameters:

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

**Method overloading:**

* It is possible only in same class.
* Static methods can be overloaded.
* Also known as static binding or early binding.
* Used to implement compile time polymorphism.
* It has same method name in same class with different signatures.
* It helps to extend functionalities.

**Method overriding**

* It is possible only in derived classes.
* The method must be a non-virtual or static method for overriding.
* Also known as dynamic binding or late binding.
* Used to implement run time polymorphism.
* Derived class has same method name with same signature as of base/parent class.
* It helps us to overwrite or change the existing functionalities.

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

* In Dart, if a class implements two interfaces, and both interfaces declare a field (variable) with the same name, there is no conflict or issue.
* Dart allows a class to implement multiple interfaces with fields of the same name without any problem.

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

* No, in Dart, a subclass instance method cannot override a superclass static method.

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

* Yes, in Dart, a subclass static method can hide a superclass instance method if they have the same name.
* This is known as method hiding.

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

* No, in object-oriented programming, a superclass cannot directly access the members of its subclass.
* The relationship between a superclass and a subclass is one-way, where the subclass inherits from the superclass, but the reverse is not true.

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

**Object-Oriented Language:**

* Object-Oriented Language supports all the feature of the OOP's.
* Java is an Object-Oriented Language as it supports all the features of the OOPs like Encapsulation, Polymorphism, Inheritance, Abstraction
* Example is C++, Java, C#, etc.

**Object-Based Language:**

* Object-Based Language does not need to support all the feature of OOP's.
* JavaScript is an Object-Based Language because you can use class and object but cannot inherit one class from another class.
* Example is JavaScript, VB, etc.