Q.21. What is inheritance?

A.21. **Inheritance** is a mechanism in which one object acquires all the properties and behaviors of a parent object.

Q.22. Which inheritance is not supported by Dart? Why? B3. What is advantage of inheritance?

A.22. Dart doesn’t support multiple inheritance.Because this inhetitance occurs when a class inherits more than one parent class. It Allows the code to be reused as many times as needed.

Q.23. Difference between inheritance and encapsulation. B5. Difference between inheritance and abstraction.

A.23**. Abstraction** is a concept that hides the implementation details and show only functionality to user.

**Inhetitance** is a concept of creationg new class using the properties and methods of an existing class.

Q.24. Difference between inheritance and polymorphism

A.24. [**Inheritance**](https://www.geeksforgeeks.org/inheritance-in-c/)**i**s one in which a new class is created that inherits the properties of the already exist class. It supports the concept of code reusability and reduces the length of the code.Types of Inheritance are:

1. Single inheritance
2. Multi-level inheritance
3. Multiple inheritances
4. Hybrid inheritance
5. Hierarchical inheritance

**Polymorphism** is that in which we can perform a task in multiple forms or ways. It is applied to the functions or methods. Polymorphism allows the object to decide which form of the function to implement at compile-time as well as run-time.

1. Compile-time polymorphism (Method overloading)
2. Run-time polymorphism (Method Overriding)

Q.25. Can we override static method in Dart?

A.25 No, We can not override static method in dart.

Q.26. Can we overload static method in Dart?

A.26. Yes, We can overload static method in dart.

Q.27. Can a class implement more than one interface? B10. Can a class extend more than one class in Dart?

A.27. Yes, a class implements more than one interface and a class can not extends more than one class.

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

A.28. Yes an interface extend more than one interface in dart with using keyword implements.

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

A.29. If two methpds have same name and signature it will give complication error.

Q.30. Can we pass an object of a subclass to a method expecting an object of the super class? B14. Are static members inherited to sub classes?

A.30. Yes, you can pass that because subclass and uperclass are related to each other by inheritance. No, Static members are not inherited to subclass.

Q.31. What happens if the parent and the child class have a field with same identifier? B16. Are constructors and initializers also inherited to sub classes?

A.31. Sub class field will hide the Super class field. Hidden super class field in sub class can be accessed using super keyword. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.

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

A.32. We can restrict a member of a class from inheriting to it's sub classes by declaring the member as a private. Because, private members are not inherited to sub classes.

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

A.33.  Dart does not support multiple implementation inheritance. Dart has interfaces, and like most other similar languages it has multiple interface inheritance.

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

A.34. A class cannot extend itself. The definition of subclass is that it extends another class and inherits the state and behaviors from that class.

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

A.35. **1)** Copy the package folder in your current code and change it as per your need.  
 **2)** Create a public method in library class and pass that private method in it. You can have the access of that private method as defined public method.

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

A.36. Overloading occurs when two or more methods in one class have the same method name but different parameters. Overriding occurs when two methods have the same method name and parameters.

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

A.37 Here my class is a class that extends’Super class’ and implements both ‘interface1’ and ‘interface2’.The ‘extends’ keyword always comes before the ‘implements’ keyword.

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

A.38. In Dart, if you want to prevent a method from being overridden without using the ‘final’ modifier, you can achieve this by defining a method in an abstract class and not providing an implemention for it. By doing so, any subclass that extends the abstract class will be required to implement the method, and they won’t be able to override it directly.

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

A.39. The method which we want to override can only be written in the subclass, not in the same class.

The return type should be the same as in the method present in the superclass.

The argument list should be the same as in the method present in the superclass.

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

A.40. The most basic difference here is that overloading is achieved in the same class, whereas overriding requires a parent and a child class at the minimum.

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

A.41. If a type implements two interfaces, and each interface define 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.

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

A.42. The static method is resolved at compile time cannot be overridden by a subclass. An instance method is resolved at runtime can be overridden.

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

A.43. 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.

Q.44. Can a superclass access subclass member?

A.44.  No, a superclass has no knowledge of its subclasses.

Q.45. Difference between object oriented and object based language.

A.45. Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects.

Examples of object-oriented programming are Java, C#, etc.

Examples of object-based languages are JavaScript, VBScript, etc.