

1.What is inheritance?

Ans:-Inheritance in programming refers to the mechanism by which a new class, called the derived class or subclass, can inherit properties, methods, and behaviors from an existing class, known as the base class or superclass. This allows the derived class to reuse code from the base class, promoting code reusability and building hierarchical relationships between classes.

The derived class inherits the attributes and behaviors (methods and fields) of the base class and can also add its own unique attributes or methods, or override existing ones from the base class. This principle is fundamental in object-oriented programming, facilitating the creation of a hierarchy of classes that share common characteristics while allowing customization and extension in more specialized classes.

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

Ans:- In Dart, multiple inheritance (where a class can inherit from more than one class) is not supported. Dart uses single inheritance, meaning a class can only inherit from one superclass. This design choice was made to simplify the language, avoid complexities related to conflicts and ambiguity arising from multiple inheritance, and promote code clarity and maintainability.

3.Difference between inheritance and encapsulation. B5. Difference between inheritance and abstraction.

Ans:-Abstraction is the methodology of hiding the internal details and displaying only the functionality to the users. In other words, the user or the external world does not see the internal implementation details. Instead, he only sees the functionality. Abstraction concept is similar to the following example. For example, assume a remote controller. It consists of a set of buttons to operate. The user can change the channels, increase the volume, change the brightness etc. He does not have to know about the internal circuitry of the controller to operate. Abstraction is similar to that.

Inheritance is the methodology of using properties and methods of an already existing class in a new class. The existing class is the parent or superclass while the new class is the child or subclass. The developer does not have to write the code from the beginning. Instead, he can use the already existing code and develop the rest. Therefore, inheritance allows code reusability.

4.Difference between inheritance and polymorphism

Ans:-Inheritance is 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 in object-oriented programming.

Types of Inheritance are:-

Single inheritance

Multi-level inheritance

Multiple inheritance

Hybrid inheritance

Hierarchical inheritance

Polymorphism:-

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.

Types of Polymorphism are:-

Compile-time polymorphism (Method overloading)

Run-time polymorphism (Method Overriding)

5.Can we override static method in Dart?

Ans:-

In Dart, static methods cannot be overridden because they are associated with the class itself rather than with instances of the class. When you declare a static method in a class, it belongs to the class and is not subject to inheritance or polymorphism.

6.Can we overload static method in Dart?

Ans:- No, Dart does not support method overloading,

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

Ans:-Implementing multiple interfaces

Dart supports only single inheritance. It means that a class can extend a single class. It doesn't support multiple inheritance, in which a class can extend multiple classes. However, a class can implement multiple interfaces.

8.Can an interface extend more than one interface in Dart?

Ans:-Implementing multiple interfaces

It doesn't support multiple inheritance, in which a class can extend multiple classes. However, a class can implement multiple interfaces. Therefore, a class can extend a single class and implements multiple interfaces.

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

Ans:-Interfaces only proscribe a method name and signature. If both interfaces have a method of exactly the same name and signature, the implementing class can implement both interface methods with a single concrete method.

10.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?

Ans:-Absolutely! In object-oriented programming, you can pass an object of a subclass to a method that expects an object of the superclass. This is known as polymorphism and is a fundamental concept in inheritance. Since a subclass inherits from its superclass, it can be treated as an instance of its superclass.

11.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?

Ans:- If both the parent and child classes have a field with the same identifier, the field in the child class will hide the field in the parent class. This means that within the scope of the child class, references to that field name will refer to the child class's field, effectively shadowing the parent class's field.

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

Ans:-You can prevent a class from being subclassed by using the final keyword in the class's declaration. Similarly, you can prevent a method from being overridden by subclasses by declaring it as a final method.

13.How do you implement multiple inheritance in Dart?

Ans:-Dart does not support multiple inheritance, but you can use mixins to achieve a similar effect. For example, you could create a Car class that inherits from a Vehicle class and a Loggable mixin. This would allow the Car class to reuse the code from both the Vehicle class and the Loggable mixin.

14.Can a class extend by itself in Dart?

Ans:-Dart allows single inheritance, which means a class can extend from a single parent. But a single parent can have multiple children, and each child can have its own children, building up a hierarchy many classes deep.

15.How do you override a private method in Dart?

Ans:-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

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

Ans:-Rules of Method overriding in Dart

The return type, list of arguments and its sequence must be the same as the parent class method. The overriding method must be defined in the subclass, not in the same class. The constructor of the superclass cannot be inherited in a subclass.

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

Ans:-All developers working with dart for application development using the Flutter framework regularly encounters different usage of the implements, extends and with keywords. In Dart, one class can inherit another class i.e dart can create a new class from an existing class. We make use of keywords to do so. In this article, we will look into 3 of the keywords used for the same purpose and compare them, namely:

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

Ans:-However, please note that as of my last update in January 2022, the @sealed annotation was not available in the stable release of Dart but was being discussed and worked on for future releases. Always check the latest Dart documentation or release notes to confirm whether this feature is available in the current version you're using.

19.What are the rules of method overriding in Dart?

Ans:-An overriding method declaration must match the method (or methods) that it overrides in several ways:

The return type must be the same type as (or a subtype of) the overridden method's return type.

Argument types must be the same type as (or a supertype of) the overridden method's argument types.

20.Difference between method overriding and overloading in Dart.

Ans:-Key Differences Between Method Overloading and Method Overriding. Following are the key differences between Method Overloading and Method Overriding. 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.

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

Ans:-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.

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

Ans:-No, we cannot override static methods because method overriding is based on dynamic binding at runtime and the static methods are bonded using static binding at compile time. So, we cannot override static methods.

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

Ans:-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.

24.Can a superclass access subclass member?

Ans:-No, a superclass has no knowledge of its subclasses. Yes, a subclass has access to all nonprivate members of its superclass.

25.Difference between object oriented and object based language.

Ans:-Object-based languages need not support inheritance or subtyping, but those that do are also termed object-oriented. Object-based languages that do not support inheritance or subtyping are usually not considered to be true object-oriented languages.