## B1. What is inheritance?

## - inheritance is defined as the process of deriving the properties and characteristics of another class. Inheritance provides reusability. With the help of inheritance we can reduce code.

## Syntax

## Class childClass extends parentClass{

## //body

## }

## B2. Which inheritance is not supported by Dart? Why?

## Dart doesn't support multiple inheritance because it creates complexity in the program. In hierarchical inheritance, two or more classes inherit a single class. In the following example, the two-child classes Peter and James inherit the Person class.

## B3. What is the advantage of inheritance?

## Benefits of Inheritance Inheritance helps in code reuse. The child class may use the code defined in the parent class without re-writing it. Inheritance can save time and effort as the main code need not be written again.

## B4. Difference between inheritance and encapsulation.

## Inheritance dictates that a child class (subclass) inherits all the attributes and methods from a parent class (superclass).

## Encapsulation dictates that one class must not have access to the (private) data of another class. The way these rules are written, it seems that they are mutually exclusive.

## B5. Difference between inheritance and abstraction.

## The main difference between abstraction and inheritance is that abstraction allows hiding the internal details and displaying only the functionality to the users, while inheritance allows using properties and methods of an already existing class. Object-Oriented Programming (OOP) is a major programming paradigm.

## B6. Difference between inheritance and polymorphism.

## Inheritance is one in which a new class is created that inherits the properties of the already existing class. It supports the concept of code reusability and reduces the length of the code in object-oriented programming. Polymorphism is that in which we can perform a task in multiple forms or ways. It is applied to functions or methods.

## B7. Can we override static methods in Dart?

## we cannot override static methods. The calling of the method depends upon the type of object that calls the static method. It means: If we call a static method by using the parent class object, the original static method will be called from the parent class.

## B8. Can we overload static methods in Dart?

## Function overloading is not supported in Dart at all. Function overloading requires static types. Dart at its core is a dynamically typed language. You can either use different names for the methods or optional named or unnamed parameters.

## B9. Can a class implement more than one interface?

## Classes can implement any number of interfaces. When you design your class, you can choose not to implement any interfaces, you can implement a single interface, or you can implement two or more interfaces. For example, in addition to IStorable, you might have a second interface, ICompressible, for files that can be compressed to save disk space.

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

## We cannot extend more than one class in Dart. If you try to extend more than one class then Dart will throw the error “ Each class definition can have at most one extended clause”. This is one of the core concepts of Object Oriented Programming in any programming language and Dart is also the same here.

## B11. Can an interface extend more than one interface in Dart?

## We cannot extend more than one class in Dart. If you try to extend more than one class then Dart will throw the error “ Each class definition can have at most one extended clause”. This is one of the core concepts of Object Oriented Programming in any programming language and Dart is also the same here. Implements:

## B12. What will happen if a class implements two interfaces and they both have a method with the same name and signature?

## This might lead to a problem: what happens if a class implements two or more interfaces that have default methods with identical method names and method signatures? The simple answer is that the implementing class must override the method. However, this must be done explicitly.

## B13. Can we pass an object of a subclass to a method expecting an object of the superclass?

## Yes, you can pass that because subclass and superclass are related to each other by Inheritance which provides IS-A property.

## B14. Are static members inherited to subclasses?

## No. Static members are not inherited. Super and subclasses, on the other hand, can both have a static method with the same signature. At the subclass, the static member of the super class will be hidden.

## B15. What happens if the parent and the child class have a field with the same identifier?

## Yes, it is possible to have the same data member in Parent and Child classes. Now, we will see the capability or strength of the Parent and Child class. The Parent class reference can hold its own object and Child class object as well and The Child class reference holds its own object only.