**Assignment -2**

**B1. What is inheritance?**

Inheritance is the capacity of a class to inherit properties and strategies from a superclass and the's superclass, etc. It is exemplified in Dart by the @override metatag. With it, a subclass's execution of inherited conduct can be particular to be proper to its more explicit subtype.

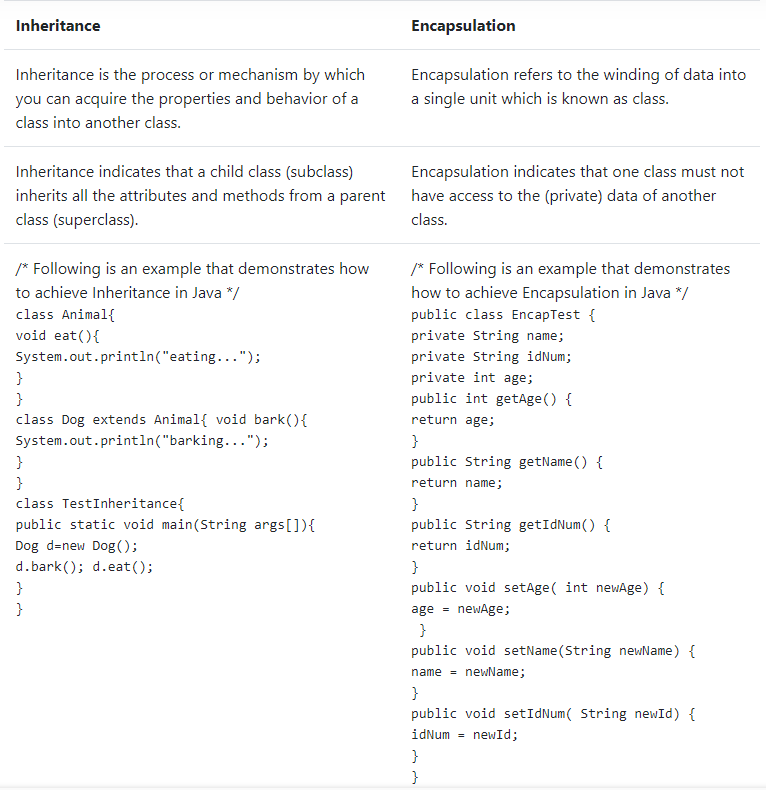
**B2. Which inheritance is not supported by Dart? Why?**

Multiple Inheritance: When a class inherits more than one parent class than this inheritance occurs. Dart doesn't support this.

**B3. What is advantage of inheritance?**

* Inheritance helps in code reuse. ...
* Inheritance can save time and effort as the main code need not be written again.
* Inheritance provides a clear model structure which is easy to understand.
* An inheritance leads to less development and maintenance costs.

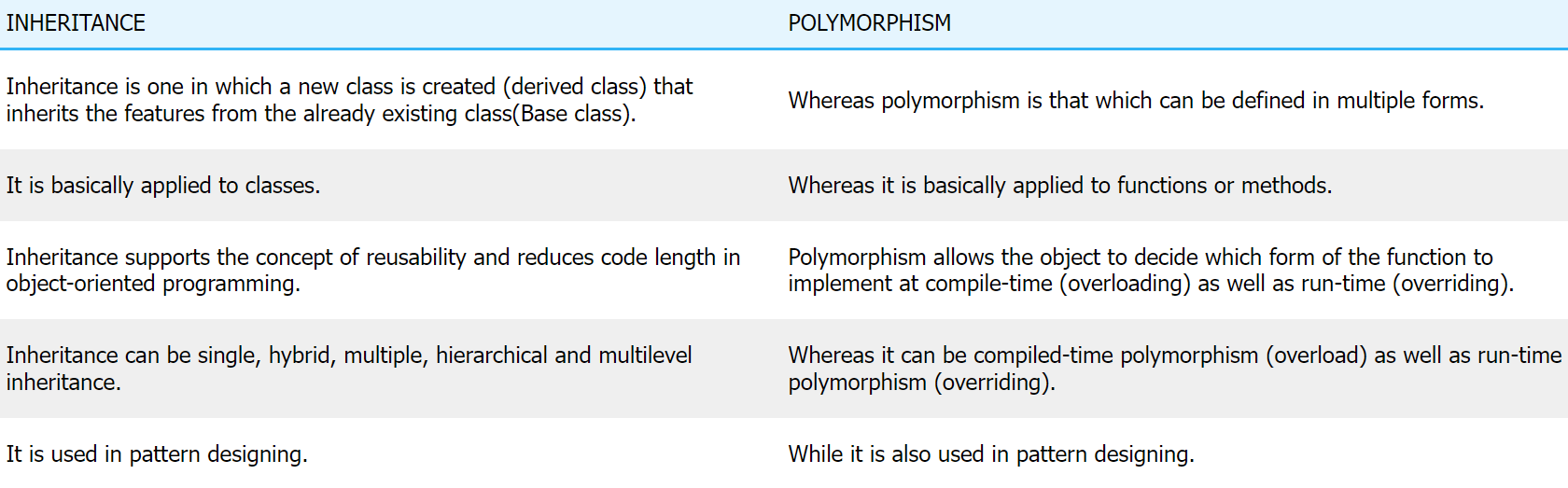
**B4. Difference between inheritance and encapsulation.**



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

|  |  |
| --- | --- |
| **Abstraction** | **inheritance** |
| OOP concept that hides the implementation details and shows only the functionality to the user | Methodology of creating a new class using the properties and methods of an existing class |
| Helps to reduce the complexity of the code | Helps to improve code reusability |

**B6. Difference between inheritance and polymorphism.**

****

**B7. Can we override static method in Dart?**

**ANS :** N**o,** You can't override static methods.

**B8. Can we overload static method in Dart?**

**ANS : yes ,** static method can be overloaded.

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

**ANS :** Yes

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

**ANS :** Yes

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

**ANS :** Yes

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

**ANS :** No, its an error ,

If two interfaces contain a method with the same signature but different return types, then it is impossible to implement both the interface simultaneously

**B13. Can we pass an object of a subclass to a method expecting an object of the super class?**

**ANS :** 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 sub classes?**

**ANS :** No. Static members cannot be inherited. However super class and the sub class can have static method with same signature.

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

**ANS :** Super class field member will be hidden at the sub class and super class field could be accessed using super keyword.

**B16. Are constructors and initializers also inherited to sub classes?**

**ANS :** No, Constructors and initializers (Static initializers and instance initializers) are not inherited to sub classes.

**B17. How do you restrict a member of a class from inheriting by its sub classes?**

**ANS :** We can restrict a member of a class from inheriting to it is sub classes by declaring the member as a private. Because, private members are not inherited to sub classes.

**B18. How do you implement multiple inheritance in Dart?**

**ANS :** No, Dart does not support multiple implementation inheritance. Dart has interfaces, and like most other similar languages it has multiple interface inheritance. For implementation, there is only a single super-class chain that a class can inherit member implementations from.

**B19. Can a class extend by itself in Dart?**

**ANS** : Unlike implementing multiple interfaces, Dart only supports single inheritance. So, you cannot extend from multiple classes.

**B20. How do you override a private method in Dart?**

**ANS :**Fundamentally you cannot override private methods of libraries. There are some alternative hacks you can go with. 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.

**B21. When to overload a method in Dart and when to override it?**

**ANS :** Method overriding occurs in dart when a child class tries to override the parent class's method. When a child class extends a parent class, it gets full access to the methods of the parent class and thus it overrides the methods of the parent class.

**B22. What is the order of extends and implements keyword on Dart class declaration?**

**ANS :** If class Second extends class First all properties, variables, methods implemented in class First are also available in Second class. Additionally, you can override methods. You use extend if you want to create a more specific version of a class.

**B23. How do you prevent overriding a Dart method without using the final modifier?**

**ANS :** The final way of preventing overriding is by using the final keyword in your method. The final keyword puts a stop to being an inheritance.

**B24. What are the rules of method overriding in Dart?**

**ANS :**

* A method can be overridden only in the child class, not in the parent class itself.
* Both the methods defined in the child and the parent class should be the exact copy, from name to argument list except the content present inside the method i.e. it can and can’t be the same.
* A method declared final or static inside the parent class can’t be overridden by the child class.
* Constructors of the parent class can’t be inherited, so they can’t be overridden by the child class.

**B25. Difference between method overriding and overloading in Dart.**

**ANS :** Method Overloading is the concept of defining two or more methods with the same name but different signatures. Method Overriding is the concept of defining two or more identical methods, having the same name and signatures.

**B26. 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.

**B27. 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.

**B28. 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.

**B29. Can a superclass access subclass member?**

**ANS :** Does a superclass have access to the members of a subclass? Does a subclass have access to the members of a superclass? No, a superclass has no knowledge of its subclasses.

**B30. Difference between object oriented and object based language.**

**ANS :** Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects, for example, JavaScript has window object. Examples for Object Oriented Languages include Java, C# whereas Object-based languages include VB etc.

**B31. Explain Diamond problem.**

**ANS :** The "diamond problem" (sometimes referred to as the "Deadly Diamond of Death") is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C.

**B32. Why Dart does not support operator overloading?**

**ANS :** By contrast, overloading would allow defining the same method or operator several times in one class, each time with different signatures (i.e. parameter lists). This is not supported by Dart, but can be emulated using dynamic or named parameters.

**B33. What is Encapsulation in Dart?**

**ANS :** Data Encapsulation is binding data and functions that use data into one unit. It is also referred to as data hiding and information hiding. Unlike C++ and Java , Dart does not have keywords for restricting access like private , public and protected .

**B34. Which of the Dart OOPS feature promotes access protection or data hiding?**

**ANS :** Encapsulation promotes access protection and data hiding.