What are the rules of method overriding in Dart?

Ans:

In Dart, method overriding is a fundamental concept in object-oriented programming that allows a subclass to provide a specific implementation of a method that is already defined in its superclass. When overriding a method in Dart, there are several rules and requirements to follow:

Inheritance: Method overriding is only applicable when a subclass inherits a method from its superclass.

Method Signature: The overriding method in the subclass must have the same name,

return type, and parameter types (method signature) as the method it is overriding in the superclass.

"@override” Annotation: It is a good practice to use the • Override annotation before

the method in the subclass that is intended to override the method in the superclass. While not strictly required, it helps catch errors at compile-time if there's a mismatch in the method signature.

class Superclass {

void method() {

print('Superclass method');

}

}

class Subclass extends Superclass {

@override

void method() {

super.method(); // Calls the superclass's method

print('Subclass method');

}

}

Superclass Method: To call the superclass's version of the method from the subclass, use the ‘super’ keyword followed by the method name.

Non-Static Methods: Overriding applies to non-static (instance) methods. Static methods cannot be overridden because they belong to the class itself and not to instances.

Constructor Overriding: Constructors cannot be overridden in the same way as regular

methods. However, you can achieve similar behavior by using factory constructors and named constructors.