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

Ans:

When a class in Dart implements two interfaces, and both of these interfaces declare a field (variable) with the same name, you may encounter a naming conflict. Dart does not support multiple inheritance, so a class can implement multiple interfaces to inherit their behavior, but it can lead to conflicts in the case of variable name clashes. Dart requires that each field in the implementing class be uniquely named to avoid ambiguity. This is part of Dart's strong type system and is intended to prevent potential issues when working with multiple interfaces.

