

Implementation and Abstraction

- **Implementation**

Implementation is used to describe the interactions of elements in programming languages. In Dart, the *implement* keyword is used to implement an interface by forcing the redefinition of the functions. For instance, every class defines an interface that contains all instance members of the class and of any interface it implements. Additionally, if you want to create some class that supports another class's API without inheriting its implementation, the class should implement its interface. Unlike inheritance, a class can implement one or more interfaces by declaring them in implements clauses and then providing the APIs required by the interfaces.

- **Abstract**

Abstraction is an essential feature of OOP, and it is used to hide background details of classes so that you only see the required information. One example of abstraction is pre-defined functions such as the *print* or *if-else* statements. Abstraction can be implemented with classes or header files that can be used publicly like “dart:io” in Dart. Users can't directly access private methods or attributes but can access the public methods in abstract classes.

- **References**

- 1- <https://www.geeksforgeeks.org/dart-extends-vs-with-vs-implements/#:~:text=The%20implement%20keyword%20is%20used,of%20any%20interfaces%20it%20implements.>
- 2- <https://www.techopedia.com/definition/23985/implementation-java>
- 3- <https://www.educative.io/answers/what-is-abstraction-in-programming>