**Interface Vs Implementation**

The separation of a class's interface against its implementation is the core principle behind encapsulation and data hiding, ensuring that only the object itself control its own actions. A class declaration combines the external interface (its non-private methods) with an implementation of that interface (the code that carries out the behavior).

The interface defines a class's visibility to others.

* Interface defines what an object can do, but doesn't do it
* Implementation carries out the operation declared in the interface
* The class design specifies the interfaces to enable an object to be instantiated and operated properly
* The Interface should completely describe how users of the class interact with the class
* The behavior of an object must be invoked by a message using one of the provided interfaces
* An object's attributes and behavior are controlled by sending messages to the object
* Messages consist of the target object's name, the desired behavior to use (i.e. the method) and any parameters required by the behavior

In programming language Interface is being used follow the encapsulation principle to hide the implementation from end user. Especially in Java language where interface type can store the object of implementation class and can be referenced to execute the methods without knowledge of implementation.

Ref:<https://oop.tech-academy.co.uk/interface-vs-implementation/>