

Difference between Abstract Class and Interface

The difference between Abstract class and Interface is a fundamental question for the java interview. Both of these are essentially used for abstraction. The level of abstraction is not same for both the processes. Interface gives 100% abstraction unlike the abstract class which cannot provide complete abstraction.

Abstract class	Interface
The implementing class can only extend one abstract class.	Allows a class to implement multiple interfaces, simulating the impact of multiple inheritances.
Methods on abstract classes can be both abstract and non-abstract.	Interfaces can only have methods that are abstract, static, or default.
The abstract class supports non-final, final, non-static, and static methods and attributes.	In the interface, only final and static methods and attributes are supported.
To extend an abstract class, use the keyword 'extend.'	The interface is implemented using the keyword implement.
Implementation Syntax: [public] class Concrete extends AbstractClass{ }	Implementation Syntax: [public] class Concrete implements Interface1, Interface2, ..., InterfaceN { }

What is Abstract Class?

An abstract class is beneficial when you want related classes to share a set of properties. Furthermore, an abstract class compensates for the limitations of defining method properties in an interface. One can define protected abstract methods/fields inside an abstract class, which interfaces cannot.

Making an abstract method private in an abstract class makes no sense. Because if an abstract method is declared private, we cannot override or access it in the abstract class's child class, which will extend the abstract class and implement its abstract methods.

What is Interface?

An interface defines the contract for the implementing class. We can think of writing down your expectations for the class to implement your interface. In other words, an

interface articulates the agreement to which you expect the implementing class to conform.

Interfaces contain a high-level description of concepts that express their disambiguated meaning whenever it is defined. An implementing class can be considered a context that clarifies the meaning of concepts defined in an interface.

