Abstract VS interface

Abstract :

•An abstract class is a class that cannot be instantiated on its own.

•It can contain a mix of abstract (unimplemented) methods and concrete (implemented) methods.

•It can also have fields (variables).

Abstract classes are used when a common base class is needed for multiple derived classes.

- They allow the definition of shared behavior and attributes among derived classes.
- Abstract classes can have abstract methods (methods without an implementation).
- Derived classes must provide implementations for all abstract methods unless they are also abstract.

Interface:

An interface is a collection of method signatures (without implementations) and constants.

It cannot be instantiated directly but can be implemented by multiple classes.

Interfaces define a contract for implementing classes. A class that implements an interface must provide concrete implementations for all methods in that interface.

Unlike abstract classes, interfaces support multiple inheritance. A class can implement multiple interfaces. All methods in an interface are implicitly abstract and public.

Starting from Java 8, interfaces can have default and static methods with implementations.

Inheritance in function constructor

In JavaScript, you can implement inheritance using function constructors and prototypes. The concept of inheritance in JavaScript is prototype-based, and it allows you to create a hierarchy of objects where a derived object inherits properties and methods from a base object.

Function constructors are functions that are used to create and initialize objects

.

They serve as blueprints for creating instances of objects with shared characteristics.

Prototype Chain:

JavaScript implements inheritance through a prototype chain.

Each object in JavaScript has an associated prototype, and when a property or method is not found on an object, JavaScript looks for it in the object's prototype chain.
Base Constructor and Derived Constructor:

A base constructor defines common properties and methods that are shared among a group of related objects. A derived constructor extends or inherits from the base constructor, adding additional properties or methods specific to the derived type.

Setting Up Inheritance:

To establish inheritance between constructors, the derived constructor's prototype is set to an object created from the base constructor's prototype.

This links the prototype chain, allowing instances of the derived constructor to access properties and methods of the base constructor.

Method Overriding:

In JavaScript, when a method is called on an object, JavaScript looks for the method in the object's prototype chain.

If the method is found in both the base and derived constructors, the method in the derived constructor overrides the one in the base constructor.