**Name:** Mina Naguib Ebead Farid

**Report**: Lap 5

**Abstract vs Interface**

An abstract class is a class that cannot be instantiated, but can contain some implementation code and define abstract methods. An abstract method is a method that is declared without any implementation, and must be overridden by subclasses.

An interface is a contract that defines the properties and methods that a class must implement, but does not provide any implementation code. A class can implement multiple interfaces, but can only inherit from one abstract class.

The main difference between abstract and interface is that abstract classes can have some concrete methods, while interfaces can only have abstract methods. Another difference is that abstract classes can have constructors, fields, and non-static or non-final variables, while interfaces cannot.

**Inheritance in Function Constructor**

In JavaScript, a function constructor is a function that is used to create objects with a specific prototype. The prototype is an object that is shared by all instances of the same constructor, and can contain properties and methods that are inherited by the instances.

To implement inheritance in function constructor:

Define the parent constructor function and add properties and methods to its prototype.

Define the child constructor function and call the parent constructor function inside it using the call () or apply () method. This will set the properties of the parent constructor to the child constructor.

Set the prototype of the child constructor to an object that inherits from the prototype of the parent constructor. You can use the Object.create () method to create such an object. This will set the methods of the parent constructor to the child constructor.

Set the constructor property of the child constructor’s prototype to the child constructor. This will ensure that the instanceof operator works correctly.