What is an abstract class?

An abstract class is a class that cannot be instantiated.

JavaScript doesn't have a native implementation of abstract classes like some other programming languages (e.g., Java or Python). However, you can achieve a similar effect using a combination of constructor functions, prototypes, and the `throw` statement.

Here's an example:

```
class Shape {
   constructor(name) {
      if(this.constructor == Shape) {
         throw new Error("Class is of abstract type and can't be instantiated");
      };
      if(this.getArea == undefined) {
          throw new Error("getArea method must be implemented");
      };
      this.name = name;
   }
}
class Rectangle extends Shape {
   constructor(name, length, width){
     super(name);
     this.length = length;
     this.width = width;
   }
}
const myShape = new Shape('My shape'); // This will throw an Error
const smallRectangle = new Rectangle("Small Rectangle", 3, 5) // This will throw an error.
```

JavaScript doesn't have built-in support for interfaces like some other programming languages do. However, you can simulate interfaces using object literals or by defining methods that must be implemented by objects.

Here's an example using an object literal:

```
// Define an interface-like object
const myInterface = {
  method1: function () {
    throw new Error("Method1 not implemented");
  },
  method2: function () {
    throw new Error("Method2 not implemented");
  }
};

// Implement the interface in a simple object
const myObject = {
  method1() {
    console.log("Method1 implemented");
  },
  method2() {
    console.log("Method2 implemented");
  }
};

// Example usage
myObject.method1(); // Outputs: Method1 implemented
myObject.method2(); // Outputs: Method2 implemented
```