INTERFACE:

In Java, an interface is an abstract class that can't be instantiated on its own and must be implemented by another class. It's a collection of abstract methods and constant variables.

Characteristics of Interfaces:

1. Abstract: Interfaces are abstract and cannot be instantiated.
2. Methods: Interfaces can only have method declarations and not method bodies.
3. Implementation: A class that implements an interface must provide an implementation for all methods declared in the interface.

Code:

```
interface Car {
    void start();
}

class ElectricCar implements Car {
    @Override
    public void start() {
        System.out.println("Electric car starts");
    }
}



public class InterfaceDemo {
    public static void main(String[] args) {
        ElectricCar tesla = new ElectricCar();

        tesla.start();

    }
}
```

EXPLANATION:

Interface Car

This interface defines a single method start(), which must be implemented by any class that implements this interface.

Class ElectricCar
This class implements the Car interface, which means it must provide an implementation for the start() method.

- The @Override annotation indicates that the start() method is overriding a method from the Car interface.
- The start() method in ElectricCar prints a message to the console indicating that an electric car has started.

Class InterfaceDemo
This class contains the main() method, which is the entry point of the program.

- An instance of ElectricCar named tesla is created.
- The start() method is called on the tesla object, which executes the implementation provided by the ElectricCar class.

Output
When the program is run, the output will be:

Electric car starts