VEHICLE HIERARCHY

SPOT

 Single & Multilevel Inheritance: Extend the existing vehicle hierarchy by adding Bike, Car, and Truck as subclasses of Vehicle. Further, create subcategories like ElectricCar and SportsCar under Car.

 Hierarchical Inheritance: Introduce another subclass AirVehicle that extends Vehicle and has subclasses Airplane and Helicopter.

 Using super:

- In the Car class constructor, call the super keyword to initialize Vehicle properties like speed and fuelType.
- Override a method in Truck and use super.methodName() to call the parent version.

 Using final:

- Declare a final method maxSpeed() in the Vehicle class that prevents overriding in subclasses.
- Create a final class HeavyVehicle to prevent further inheritance.

E-COMMERCE PLATFORM

SPOT

 Single & Multilevel Inheritance: Create a hierarchy where Product is the base class and Electronics, Clothing, and Furniture are subclasses. Further extend Electronics into Mobile and Laptop.

 Hierarchical Inheritance: Introduce DigitalProduct as another subclass of Product, which is extended by EBook and OnlineCourse.

 Using super:

- Call super in Electronics to set product attributes inherited from Product.
- Use super.methodName() to reuse a method in Clothing.

 Using final:

- Mark final a method generateInvoice() to prevent modifications in subclasses.
- Declare final class OrderIDGenerator that ensures unique order IDs.

CAR RENTAL SYSTEM

SPOT

 Single & Multilevel Inheritance: Define a Car base class and extend it into EconomyCar, SUV, and LuxuryCar. Further, extend LuxuryCar into SportsCar.

 Hierarchical Inheritance: Have Truck as another subclass of Vehicle, which is extended by CargoTruck and DeliveryVan.

 Using super:

- In the LuxuryCar constructor, use super to initialize the base Car attributes.
- Override calculateRentalCost() in EconomyCar and SUV while using super.calculateRentalCost() to reuse logic.

 Using final:

- Make final the method getRentalTerms() to enforce standard policies.
- Declare final class RentalAgreement to prevent modifications.

RIDE-SHARING SYSTEM

SPOT

 Single & Multilevel Inheritance: Define a Ride base class and extend it into EconomyRide, LuxuryRide, and SharedRide. Further extend LuxuryRide into VIPRide.

 Hierarchical Inheritance: Have a separate Vehicle class that is extended by Bike and Car.

 Using super:

- In the LuxuryRide constructor, call super to initialize Ride attributes like driver, passenger, and fare.
- Override calculateFare() and use super.calculateFare() to reuse base logic.

 Using final:

- Mark final the method getBaseFare() to enforce fare rules.
- Declare final class RideIDGenerator to ensure unique ride IDs.

SMART HOME AUTOMATION

SPOT

 Single & Multilevel Inheritance: Define a SmartDevice base class and extend it into Light, Fan, and AirConditioner. Further extend AirConditioner into SmartAC.

 Hierarchical Inheritance: Introduce SecurityDevice as another subclass of SmartDevice, extended by SmartLock and SurveillanceCamera.

 Using super:

- Call super in SmartAC to initialize SmartDevice properties like deviceID and powerConsumption.
- Override turnOn() and use super.turnOn() to reuse base logic.

 Using final:

- Mark final a method deviceDiagnostics() to enforce standard behavior.
- Declare final class HomeIDGenerator to generate unique home IDs.