

```

public class Car extends Vehicle implements Rentable {
    private boolean hasGPS;

    public Car(String vehicleId, String model, double baseRentalRate, boolean hasGPS) {
        super(vehicleId, model, baseRentalRate);
        this.hasGPS = hasGPS;
    }

    public boolean hasGPS() {
        return hasGPS;
    }

    public void setHasGPS(boolean hasGPS) {
        this.hasGPS = hasGPS;
    }

    @Override
    public double calculateRentalCost(int days) {
        double cost = getBaseRentalRate() * days;
        if (hasGPS) cost += 50; // GPS adds an extra flat fee
        return cost;
    }

    @Override
    public boolean isAvailableForRental() {
        return isAvailable();
    }

    @Override
    public void rent(Customer customer, int days) {
        if (!isAvailableForRental()) throw new IllegalStateException("Car is not available.");
        setAvailable(false);
        System.out.println("Car rented by " + customer.getName() + " for " + days + " days.");
    }

    @Override
    public void returnVehicle() {
        setAvailable(true);
        System.out.println("Car returned.");
    }
}

```

