

```

public class Truck extends Vehicle implements Rentable {
    private double loadCapacity;

    public Truck(String vehicleId, String model, double baseRentalRate, double loadCapacity)
    {
        super(vehicleId, model, baseRentalRate);
        this.loadCapacity = loadCapacity;
    }

    public double getLoadCapacity() {
        return loadCapacity;
    }

    public void setLoadCapacity(double loadCapacity) {
        this.loadCapacity = loadCapacity;
    }

    @Override
    public double calculateRentalCost(int days) {
        return getBaseRentalRate() * days + (loadCapacity * 0.5);
    }

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

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

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

```

