```
public abstract class Vehicle {
   // Private encapsulated fields
   private final String vehicleId;
   private String model;
   private double baseRentalRate;
   private boolean isAvailable;

   // Constructor with validation
   public Vehicle(String vehicleId, String model, double baseRentalRate) {
      if (vehicleId == null || vehicleId.isEmpty()) throw new IllegalArgumentException("Vehicle
ID cannot be null or empty.");
       if (model == null || model.isEmpty()) throw new IllegalArgumentException("Model
cannot be null or empty.");
      if (baseRentalRate <= 0) throw new IllegalArgumentException("Base rental rate must be
positive.");

       this.vehicleId = vehicleId;
       this.model = model;
       this.baseRentalRate = baseRentalRate;
       this.isAvailable = true;
   }

   // Abstract methods for rental calculation and availability
   public abstract double calculateRentalCost(int days);
   public abstract boolean isAvailableForRental();

   // Getters and Setters
   public String getVehicleId() {
      return vehicleId;
   }

   public String getModel() {
      return model;
   }

   public void setModel(String model) {
      if (model == null || model.isEmpty()) throw new IllegalArgumentException("Model
cannot be null or empty.");
      this.model = model;
   }

   public double getBaseRentalRate() {
```

```
        return baseRentalRate;
    }

    public void setBaseRentalRate(double baseRentalRate) {
        if (baseRentalRate <= 0) throw new IllegalArgumentException("Base rental rate must be
positive.");
        this.baseRentalRate = baseRentalRate;
    }

    public boolean isAvailable() {
        return isAvailable;
    }

    public void setAvailable(boolean available) {
        isAvailable = available;
    }

    @Override
    public String toString() {
        return "Vehicle{" +
                "vehicleId='" + vehicleId + '\'' +
                ", model='" + model + '\'' +
                ", baseRentalRate=" + baseRentalRate +
                ", isAvailable=" + isAvailable +
                '}';
    }
}
```