The main driver of the program is the Application.java file.

Each pattern has its own test drive method:

- IteratorPatternTestDrive.java
- AbstractFactoryPatternTestDrive.java
- DecoratorPatternTestDrive.java
- ObserverPatternTestDrive.java

Each class has a comment on the top stating which pattern it belongs too. The complete left is below, and the reflection is at the end of the document.

There are 68 files to the program that I was unable to package separately but most of them are abstract factory and decorator components. The main driver is ~360 lines of code but it's half that because the code is repeated for car and SUV creators and decorators repeats in the loop.

Iterator Pattern Files:

- AddCarInventory
- AddTruckInventory
- Automobile
- CarInventory
- CarInventoryIterator
- InventoryPrinter
- Iterator
- TruckInventory
- TruckInventoryIterator
- IteratorPatternTestDrive

Observer Pattern Files:

- Subject
- ObserverDisplay
- Observer
- Magazine
- LotData
- ObserverPatternTestDrive

Abstract Factory Pattern Files:

- AbstractFactoryPatternTestDrive
- BaseCarFactory
- BaseCarStore
- BaseSUVFactory
- BaseSUVStore
- EliteCarFactory
- EliteCarStore
- EliteSUVFactory

- EliteSUVStore
- LuxuryCarFactory
- LuxuryCarStore
- LuxurySUVFactory
- LuxurySUVStore
- Seat
- ClothSeat
- LeatherSeat
- LeatherHeatedSeat
- CarVehicle
- Engine
- FourCylinderEngine
- SixCylinderEngine
- NineteenInchRim
- NoSunroof
- SeventeenInchRim
- Sunroof
- SUVVehicle
- TwentyInchRim
- YesSunroof
- Vehicle
- VehicleManufacturerFactory
- VehicleStore
- YesSunroof
-

Decorator Pattern Files:

- BikeRack
- BodyKit
- BoseAudio
- Cameras
- CarbonFiberAccents
- DecoratorPatternTestDrive
- Exterior
- FeatureDecorator
- HeatedSteeringWheel
- Interior
- LeatherGearShift
- LEDLights
- Navigation
- Sensors
- Technology
- Tints
- VehicleFeatures
- Wireless

Reflection

Overall, I did not find the project too difficult as I felt the homework assignments and github code provided for the book did a good job helping me build a template that I can use to quickly implement the different design patterns. While studying for this course, I build class maps with the code provided for the book that helped me understand how to use these patterns quickly and understand their general layout, so coding wasn't too difficult.

The most difficult part of the project for me was how to set up my main driver function. In the loop, I had to figure how to make the program flow in a way that made sense and where to initialize my objects. To overcome this difficulty, I tried to think of the process of building and adding features to a car from the viewpoint of a consumer. A second difficult, I would say, was to decide which design patterns to use for the different functionalities of the program. To help me overcome this difficulty, I tried to imagine the process of buying a car again. For the building of a vehicle from a manufacturer, the builds for each car are predefined so a abstract factory pattern made the most sense. To add custom features to a car, such as body kits and sensors, the manufacturer doesn't necessarily need to install them so a decorator pattern where you can pile additional features made the most sense to me. The observer pattern logically makes the most sense for when a new vehicle is available for sale, its best to notify everyone who is subscribed to look at vehicles. The iterator pattern makes sense to go over and print all the vehicles in the current inventory with price like one would a diner menu (as shown in class).

I tried to keep the code as clean as possible and did not make it very robust due to the already large size so there are bugs present. For example, one bug is that when you create multiple cars or SUVs, the features decorators carry over the previous build. The source file is also messy to look at and I tried to package the code, but I was getting errors as I moved files into the folder. Given the approaching deadline, I couldn't overcome this difficulty how I wanted and added comments to my code along with pattern files in this document. I hope this helps with your readability and apologize about the size of the project, before I knew what happened, it kind of spiraled out of control.

I really enjoyed this class and found your lectures and insights helpful. Your efforts demonstrate how much you care about your students and if they are able to learn the material. You also have an obnoxiously large number of students and didn't lose attention to detail when reviewing over 100 assignments a week and left comments where something wasn't implemented correctly. I appreciate your efforts and having an instructor like yourself is a great experience.

Thank you for a great class and learning experience!