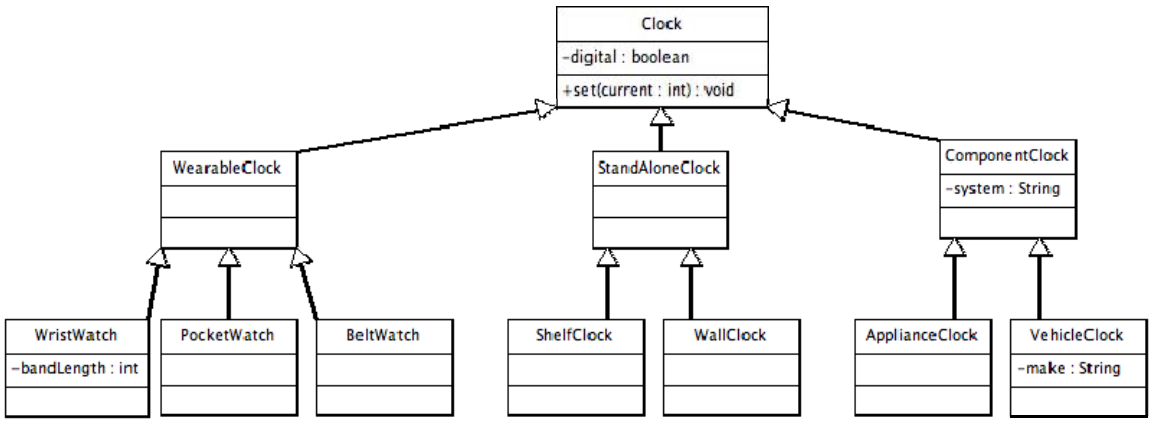
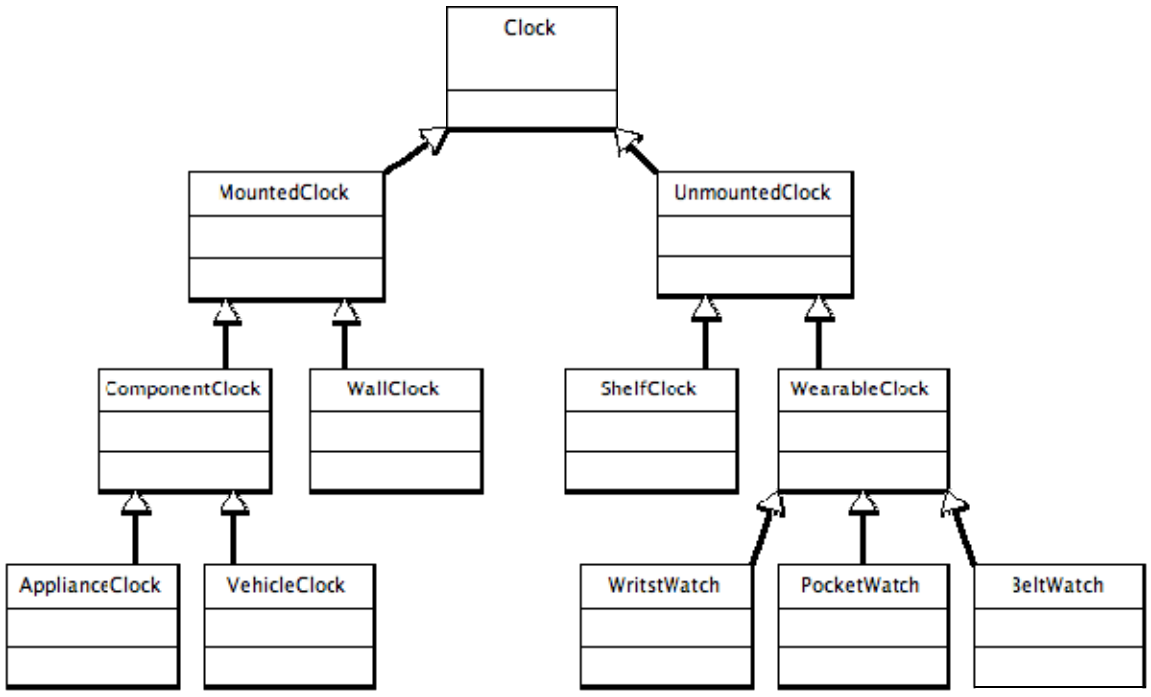
Chapter 8 Exercise Solutions

EX 8.1. Draw a UML class diagram showing an inheritance hierarchy containing classes that represent different types of clocks. Show the variables and method names for two of these classes.

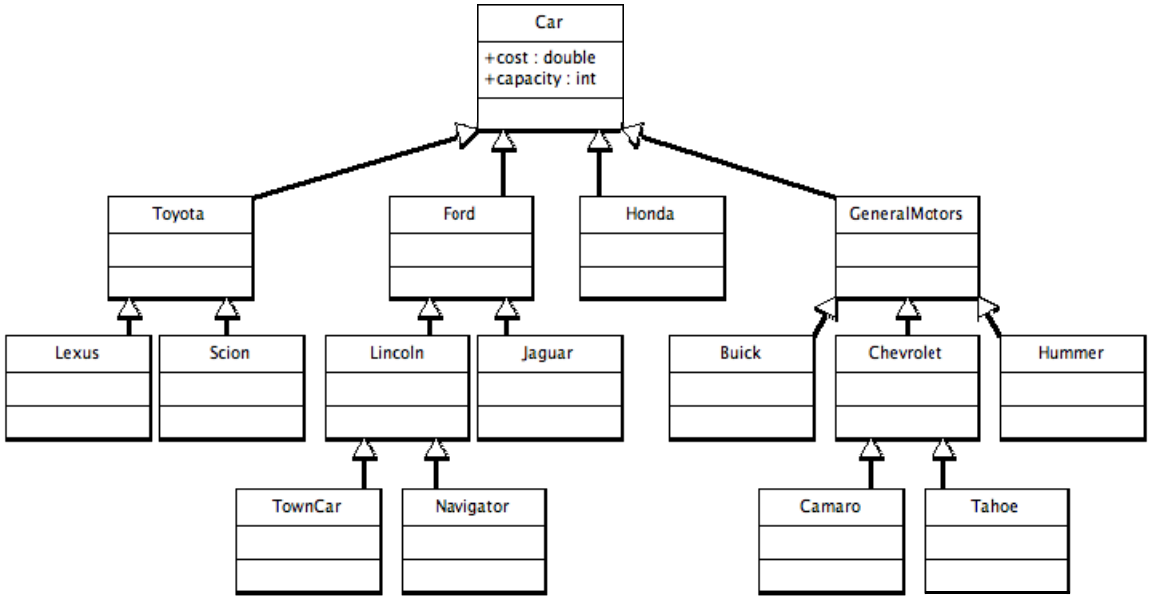


EX 8.2. Show an alternative diagrm for the hierarchy in Exercise 8.1. Explain why it may be a better or worse approach than the original.

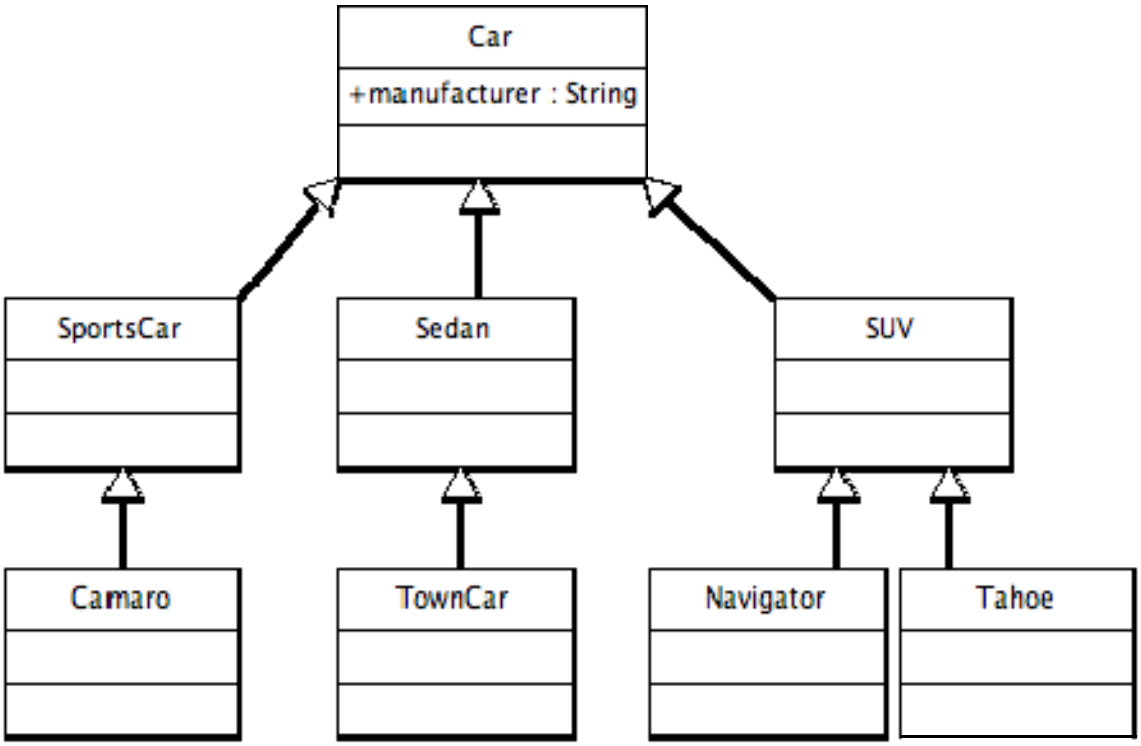


The value of the organization is dependent on how the classes are used. The hierarchy in Ex. 8.1 might be better suited to a system maintaining the inventory and online shopping for a store, whereas the hierarchy in 8.2, centered around the mounting capabilities, might be better suited to a manufacturing system.

EX 8.3. Draw a UML class diagram showing an inheritance hierarchy containing classes that represent different types of cars, organized first by manufacturer. Show some appropriate variables and method names for at least two of these classes.

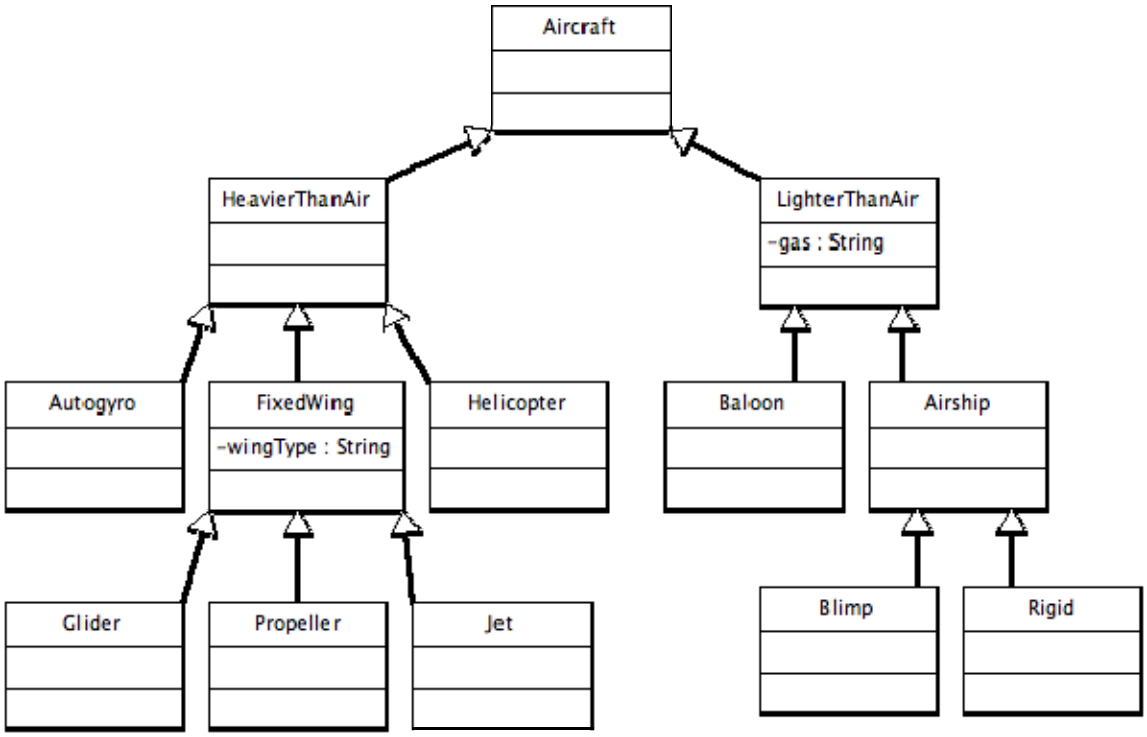


EX 8.4. Show an alternative diagram for the hierarchy in Exercise 8.3 in which the cars are organized first by type (sports car, sedan, SUV, etc.). Show some appropriate variables and method names for at least two of these classes. Compare and contrast the two approaches.

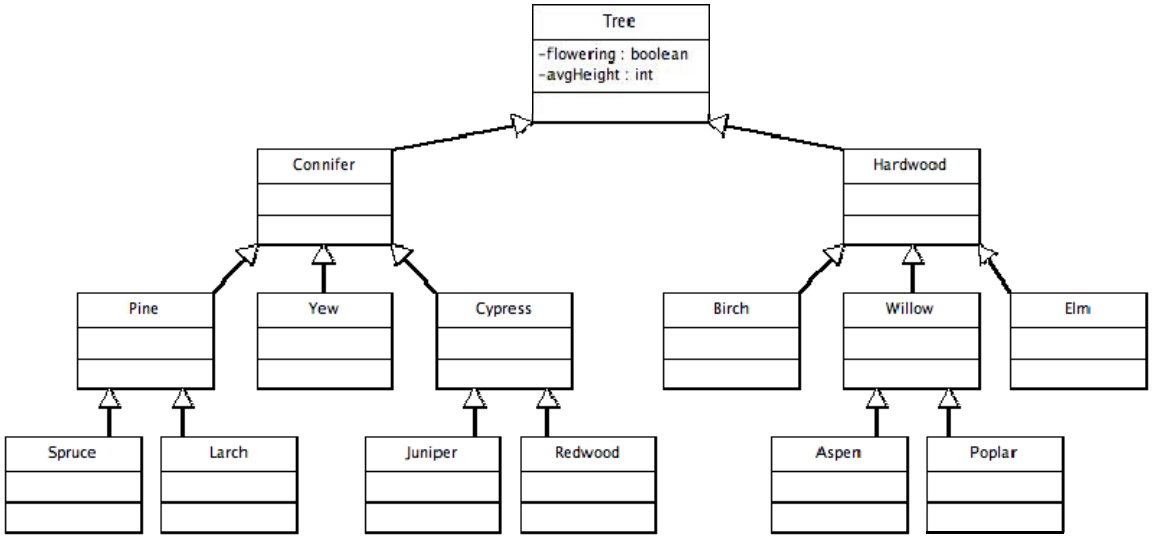


The hierarchy in Ex. 8.3 has the manufacturer information as an inherent part of the inheritance structure, and might store car type information as data in lower level classes. The hierarchy in Ex. 8.4 is organized around car type, with the manufacturer as data. Which one would be best depends on the purpose of the system.

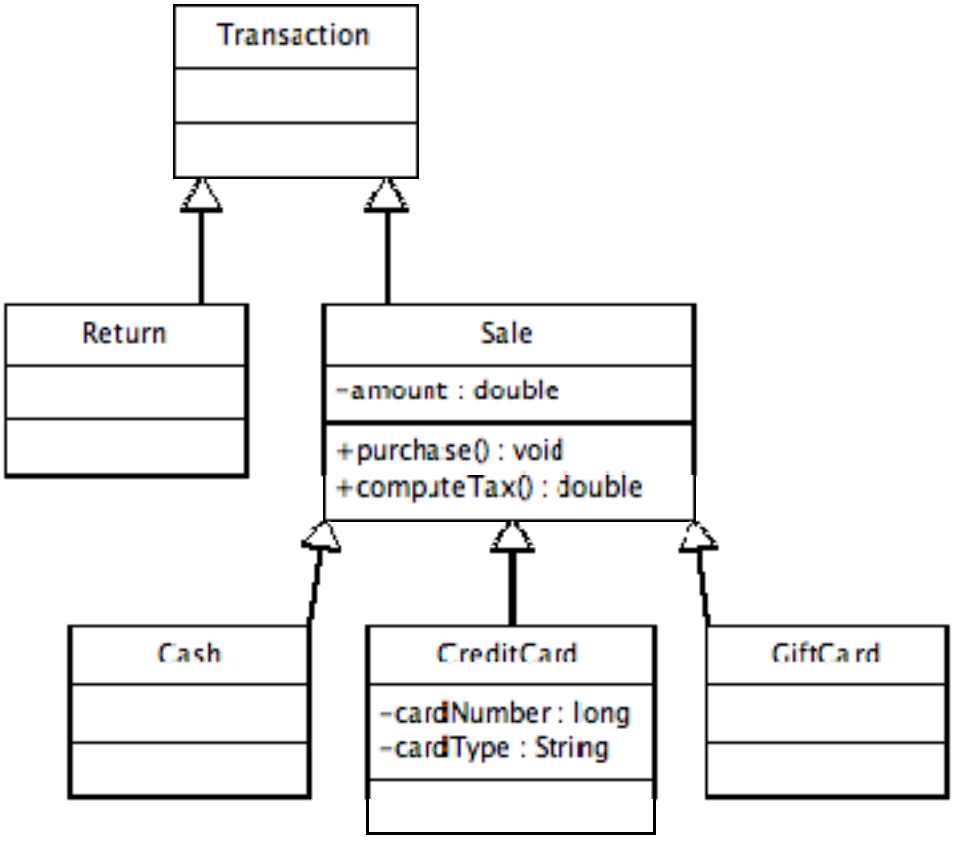
EX 8.5. Draw a UML class diagram showing an inheritance hierarchy containing classes that represent different types of airplanes. Show some appropriate variables and method names for at least two of these classes.



EX 8.6. Draw a UML class diagram showing an inheritance hierarchy containing classes that represent different types of trees (oak, elm, etc.). Show some appropriate variables and method names for at least two of these classes.



EX 8.7. Draw a UML class diagram showing an inheritance hierarchy containing classes that represent different types of transactions at a store (cash, credit card, etc.). Show some appropriate variables and method names for at least two of these classes.



EX 8.8. Experiment with a simple derivation relationship between two classes. Put println statements in constructors of both the parent and child classes. Do not explicitly call the constructor of the parent in the child. What happens? Why?

Change the child's constructor to explicitly call the constructor of the parent. Now what happens?

When a parent's constructor is explicitly called, its program statements are executed as expected. But even in the absence of an explicit call, the parent's constructor is called when the child class is instantiated. An explicit call allows the child to pass parameters, allowing the parent to set up its contribution to the child's state. Without an explicit call, the default constructor is called, without parameters.