<these are from chapter 6 though>

5.1 What's the difference between a component–based architecture and a service–oriented architecture?

Component–based architecture splits the program to a number of pieces, where the pieces are stored in the same program and can access each other directly.

Service–oriented architecture is a collection of multiple programs, and the pieces could communicate with each other internet.

5.2 Suppose you're building a phone application that lets you play tic–tac–toe against a simple computer opponent. It will display high scores stored on the phone, not in an external database. Which architectures would be most appropriate and why?

I would use a monolithic architecture, because tic–tac–toe does not sounds like a big program, and the data is stored within the phone, so this program does not use any outside database to store, so a monolithic architecture will just do.

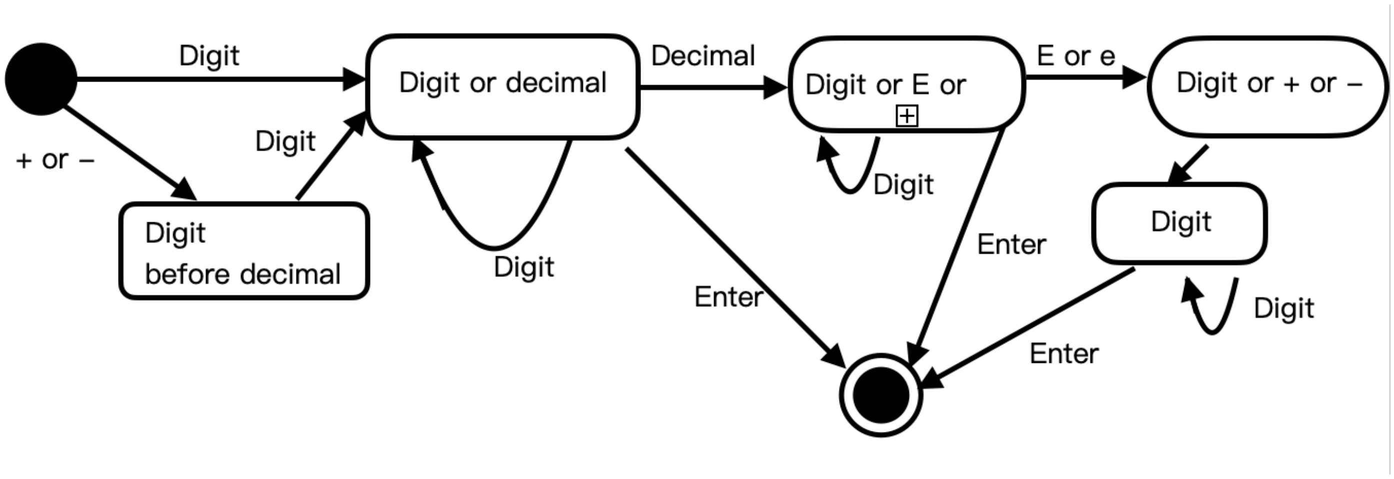
5.4 Repeat question 3 [after thinking about it; it repeats question 2 for a chess game] assuming the chess program lets two users play against each other over an Internet connection.

This should be a client–server architecture, because it have multiple user interface and they communicate through network. The user side will give commands, and the database side will receive it and decide if it is doable, and send the result back to the two users.

5.6 What kind of database structure and maintenance should the `ClassyDraw` application use?

It should be a document database, because each component drew is an independent element. Backup and recovery would be the most important function in maintenance, because loosing progress for drawing is extremely annoying.

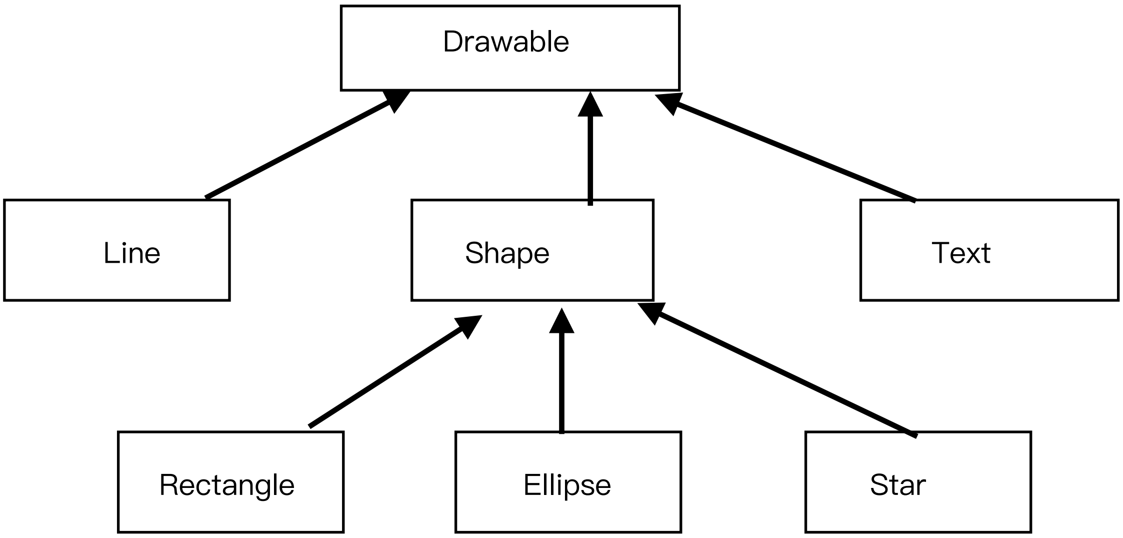
5.8 Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +37 or –12.3e+17 (which means –12.3 x 1017). Allow both E and e for the exponent symbol. [Jeez, is this like Dr. Dorin's DFAs, or what???]



6.1 Consider the `ClassyDraw` classes `Line`, `Rectangle`, `Ellipse`, `Star`, and `Text`. What properties do these classes all share? What properties do they not share? Are there any properties shared by some classes and not others? Where should the shared and nonshared properties be implemented?

They should all be objects that can be created, selected, and changed around. Text should have the sole property of being able to modify the content of it. Line should allow user to choose where the two ends should be. Rectangle, Ellipse and Star can share the property of being a shape, though I'm not sure what special properties do they have that separate them from Line and Text.

6.2 Draw an inheritance diagram showing the properties you identified for Exercise 1. (Create parent classes as needed, and don't forget the `Drawable` class at the top.)


Drawable
Line
Shape
Text
Rectangle
Ellipse
Star