1. What's the difference between a component-based architecture and a service-oriented architecture?

A component-based architecture is different from a service-oriented architecture in that it breaks down an application into modules that communicate through interfaces in the same codebase. In a service-oriented architecture, each major function is an independent service that runs in its own process, which is usually done over a network.

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?

For mobile tic tac toe application, a monolithic architecture is most appropriate because you only need the UI and game logic to run on the phone, removing the need for distributed structures.

3. 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.

For a chess program that lets users play against each other over an internet connection, a client/server architecture is required, where each phone acts as a client while the server manages the shared game state and updates the UI for both players.

4. What kind of database structure and maintenance should the ClassyDraw application use?

Since ClassyDraw seems to be a local program, files created and modified in this app can probably be saved to local disk. The app should periodically auto-save files and be able to load previously edited files within the editor.

5. 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. Consider the ClassyDraw classes Line, Rectangle, Ellipse, Star, and Text.
    a. What properties do these classes all share?

These classes all share properties like position and stroke color.

    b. What properties do they NOT share?

They do not share class-specific properties like fill color or font across all classes.

    c. Are there any properties shared by some classes and not others?

Some classes that represent shapes, such as rectangle, ellipse, and star share the property of fill while other classes like line and text do not need a fill property.

    d. Where should the shared and nonshared properties be implemented?

Shared properties should exist in the parent Drawable class, while the class-unique properties should exist in their respective subclasses.

7. Draw an inheritance diagram showing the properties you identified for Exercise 6.1. [Create parent classes as needed, and don't forget the Drawable class at the top.]