Problem 5.1

Component-based architecture uses loosely coupled components within a single program that provide services to each other through direct communication. Service-oriented architecture uses self-contained services that run independently on separate systems and communicate over networks using protocols like SOAP or REST.

Problem 5.2

Monolithic architecture is most appropriate. The game is simple with no database, no network, and runs entirely on one device. All functionality (UI, game logic, score storage) can exist in a single program without needing separation into tiers or services.

Problem 5.4

Client/server or service-oriented architecture. Two users playing over the Internet requires network communication between separate devices. One device acts as the server managing game state while both devices act as clients displaying moves and accepting input.

Problem 5.6

ClassyDraw should use file-based storage, not a database. Drawings are stored as files that users can save and load. No complex queries, multiple users, or relational data is needed. Simple file I/O is sufficient for this standalone drawing application.

Problem 5.8

Starting from the Start state, the machine reads optional +/-, then digits, then optional decimal point and more digits, ending with Enter. Valid path: Start → (+/-) → digits → (decimal) → digits → Enter → Stop.

Problem 6.1

1. All classes share position, line style, line color, selected state, and a draw method.
2. They don't all share fill properties, font properties, number of points, or text content.
3. Rectangle, Ellipse, and Star share fill style and fill color, but Line and Text don't. Only Text has font properties. Only Star has point count.
4. Shared properties go in a Drawable base class. Fill properties go in a FilledShape class that inherits from Drawable. Text properties stay in Text class only. Line inherits directly from Drawable. Rectangle, Ellipse, and Star inherit from FilledShape.

Problem 6.2