

# Software Engineering

CSE470

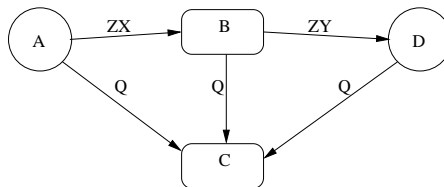
Spring 2002

Homework 1

Due: 1/24/2002

1. An *abstract class* is class from which no objects are instantiated and serves as a common parent in an inheritance hierarchy. For example, a **polygon** abstract class could be the superclass for a **triangle** and **square** class. Only **squares** and **triangles** would be instantiated, but all the common attributes would be in **polygon**. You may find abstract classes useful in this problem.

We want to design a drawing tool for simple finite state machines. A *machine* is a collection of one or more *states* and zero or more *transitions*. Our customers can't decide whether they want circular states or rectangular states with rounded corners, each of which has it's own drawing format, so we'll allow both forms in our tool. Each state can have zero or more transitions outbound, and zero or more transitions inbound. States have an  $x$  and  $y$  location on the canvas and a name string. Transitions have a string that denotes the transition expression. Draw the class diagram to represent the figures used in this tool (state machine, states (both kind), and transitions). Be sure to get the multiplicities correct, show all relevant associations, and show the attributes mentioned above. Here is an example of one of these goofy diagrams:



Note: You may hand-draw the diagram. If you do, use the  $\text{\LaTeX}$  `\vspace` command to make white space. For example, `\vspace{2in}` reserves two inches of space on the page.

2. Describe the difference between *generalization* and *abstraction*. Don't simply define each.