CSE 403 - Software Engineering
Notes on Homework #1
January 28, 1997

Problem #1 Credit Card System DFD

- Arrow Labeling
  - All arrows should be labeled with the data that is flowing along the arrow.
  - The arrow labels should just be the *data*, not actions on the data.
  - There should not be any conditional statements on the arrows.
- Inputs
  - Inputs should represent where the data originates.
  - Examples: Salesclerk, Customer, Register
  - No data should flow INTO an Input, only out of the input.
- Data Stores
  - There can be no operations on the data within the data store (that is what functions are for), so the data that goes into a data store should be the same as that coming out.

Problem #2 Finite State Machines

- Part A
  - The outputs should be labeled on the transitions with the inputs (i.e. *input/output*), not a separate states.
  - Once more than $0.50 has been entered, you should immediately return all money entered.
  - You can diagram the returning of change one of two ways:

        * have multiple "returning change" states that require no input to output the appropriate coin, or
        * label your transitions with multiple outputs (then you can return all the necessary coins at once)

- Part B
    - This diagram should only require 2 or 3 states since states are no longer needed to save information about how much money has been collected.
    - You need to have gaurds on accepting all quarters and dimes since you do not want to accept them if you have already collected $0.50.

Problem #3 Railroad Crossing State Machine

- There should be a State Machine to diagram the behavior of each system in the crossing: the controller, the gates and the light. The diagrams should show the state transitions each of these systems goes through independently.
- The Controller's output is input to the Gates and Lights. The Gate's output is input to the Lights. Although these systems depend on each other for input, they are still in parallel since the actions within the systems do not sequentially rely on each other (they are all functioning in parallel).
- The Gate state machines should have states for the gate opening and closing since it takes some time to get from the open to closed position and vice versa.
- The light can turn to green only when the gate is completely open. The input for the lights to change to green can only come from the Gates once the gate has reached the *open* state.

Problem #4 VCR State Machines

I accepted just about any reasonable model of how a VCR works. You needed to show some interrupting behavior between different modes. The example in the solution set is just one way to do it. There were a large number of correct solutions.