**CS431, Fall 2006**
**Prof Barbara Ryder**
**Individual Graded Assignment 2: Domain Modeling and Design**
**Posted: Tues., October 17, 2006**
**Due: midnight on Fri., October 27, 2006**

1. This domain model represents orders and customers that place these orders. The model has problems. (a) Briefly explain at least 2 of the problems. (b) Show an improved domain model that does not have these problems.

```
┌─────────────────────┐        ┌─────────────────────┐
│        Order        │        │      Customer       │
├─────────────────────┤        ├─────────────────────┤
│      date:Date      │        │    name: String     │
│    price:Integer    │        │   address:Address   │
│  customer:Customer  │        └─────────────────────┘
└─────────────────────┘
```

2. Consider a software system that processes course information in a university. The system has to represent courses and classes. Each *course* has an ID (e.g. "CSE 757") and a string title (e.g. "Software Engineering"). Each *class* is an offering of a particular course for a particular semester (e.g. "Fall 2006"), and has its own call number (e.g. "04953-1"). Assume that classes are not cross-listed in different departments under different course IDs (i.e., assume that each class is related to exactly one course and course ID).

(a) Show a domain model for these concepts, with all relevant classes, attributes, associations, and multiplicities implied by the above description. Do not show elements that are irrelevant to the description.

(b) Extend the domain model from (a) to represent the fact that students can register for classes. Do not replicate the entire model from (a); just show the new elements and describe how they fit into the existing model.

(c) Suppose that when a student registers for a class, she can choose to register at one of two commitment levels: *tentative* or *certain*. After registering, a student may choose to change the level (e.g., to change from *tentative* to *certain*, or vice versa) at any point of time; she may even change levels many times. Extend the model from (b) to represent this level. Do not replicate the entire model from (b), just show the new elements and describe how they fit into the existing model.

3. Consider a software system that stores information about students and their current status in a university (i.e., freshman, sophomore, junior, senior) using the domain model shown below. Discuss briefly whether or not this is a good domain model. To make this more concrete, state specific tasks for which the structure is suitable or not suitable.

```
                    ┌──────────────┐
                    │   Student    │
                    └──────△───────┘
                           │
        ┌──────────┬───────┴───────┬──────────┐
   ┌─────────┐┌──────────┐   ┌─────────┐ ┌─────────┐
   │Freshman ││Sophomore │   │ Junior  │ │ Senior  │
   └─────────┘└──────────┘   └─────────┘ └─────────┘
```