

CS 249 MidTerm Exam - Closed Book In-Class Exam

Time allotted: 75 minutes November 4, 2003

Answer all questions. Each question is worth the points as indicated. You can answer in point form if you prefer when English is called for. You can use sketch code that is not necessarily complete AS LONG AS IT CLEARLY INDICATES you understand the approach or technique we are after. We are not looking for a full essay on each question, but rather a short concise set of points or sketch code that responds to the question and indicates you understand the point being explored.

- (15 Points)** Cheriton praises the advantages of the SOS approach yet it has some significant "challenges", modern management speak for difficulties that you need to conquer or else die in battle. Describe, for each of the following aspects of SOS, the key challenge and how we try to conquer it.
 - source code representation
 - outside-in development
 - short cycle development
- (15 Points)** Describe the semantics that are required for each of the following according to the course.
 - read accessor
 - write accessor
 - destructor
- (15 Points)** You talk your software team into swallowing the attribute-only story hook, line and sinker. However, then your boss comes along and says the network server you are working on needs to support hot-swapping of line cards/blades. Suddenly, there is a revolt from your team: "clearly we need to use verbs to say *insert* and *remove* to support hot-swapping, i.e. the inserting and removing of cards.", says your team.
 - Describe the attribute-only interface(s) you would use to provide this functionality.
 - Justify why this is better than the "verby" approach. (Or you can describe a verb solution and defend that, if you can make a good case.)
- (15 Points)** Describe the key differences between the 3 categories of types Cheriton identifies, i.e. value, entity and named description types.
- (12 Points)** Various stock trading systems have used the publish-subscribe model to implement stock tracking and monitoring facilities. Basically, the stocks and stock indexes are modeled as "publishers". The traders then "subscribe" to updates on changes to stocks and stock indices of interest. The course reader describes how you can view the notification model as a means to implement publish-subscribe within an object-oriented structure. Starting with the `StockMarket` class below, specify the attributes, subclasses and definition of `StockMarket::Notifieee` needed for such a pub-sub system within our notification model.

```
class StockMarket {
    class Notifieee;
    . . .
};
```

Describe just the client-visible interfaces and sketch briefly how it would work.

- 12 points** Considering exceptions as explored in the course:

- (a) Stroustrup worked hard to allow you to define arbitrary exception types and still be able to easily catch all possible exceptions by having them derived from a common root `exception` class. Nevertheless, Cheriton wants to restrict interface exception types to just resource exceptions. Describe the reasoning behind this restriction.
- (b) Cheriton makes a big deal of restricting whole categories of functions from throwing exceptions, arguing it is important for achieving the attribute semantics of Chapter 2. Describe why these restrictions are important in implementing the transactional aspect of attribute semantics, with suitable reference to the restricted categories. (Do not describe any other reasons for these restrictions.)

The End