

CS 249 MidTerm Exam - Closed Book In-Class Exam

Time allotted: 75 minutes November 4, 2004

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.

1. **(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 three key approaches to dealing with the demands of source code representation of the design.
2. **(15 Points)** In Cheriton's attribute-only interface model, a read accessor is exception-free, nilpotent and transactional. Give an example of a demanding aspect required of the programmer for each of these.
3. **(15 Points)** A senior technical lead in your project claims that Cheriton's classification of types into value, entity and named descriptions might be nice academic thing, but is really pretty arbitrary and has no practical implications. Describe three key situations which support this classification and the associated restrictions with each.
4. **(15 Points)** Consider implementing a `Broker` module that maintains a collection of `StockIndex`'s, which it monitors using the `StockIndex::Notifieee` interface, declared as:

```
class StockIndex ; public NamedInterface {
    class Notifieee;
    . . .
};
```

Draw a picture illustrating as clearly as possible the modules involved, which one implements each interface, the different data structures and the path through these data structures that a notification takes.

5. **15 points** Suppose you join a software company that is just about to starting using C++ exceptions and Fred, one of your new co-workers, argues that each programmer should define his/her own module-specific exception types. Describe the implications to taking Fred's approach as opposed to the approach advocated by this course.

The End