

CS 249 MidTerm Exam - Closed Book In-Class Exam

Time allotted: 75 minutes November 10, 2005

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)** A key challenge for large-scale software is reducing the size of the "slice" in a mountain of software that an engineer has to learn to become useful. Describe three techniques or approaches that have been presented in the course to date that are targeted to this goal, and briefly sketch how they achieve this.
2. **(15 Points)** In Cheriton's attribute-only approach, conventional wisdom about separating interface from implementation is violated by using inline accessors and data members in the interface class. Describe the degree to which this compromises the benefits of separating interface from implementation versus the benefits of the attribute approach.
3. **(15 Points)** Give an example of a convention used with value types that is not used with entity types, and vice versa, justifying why in both cases.
4. **(15 Points)** You are given a `Sensor` interface that supports notification, i.e.

```
class Sensor : public NamedInterface {
    class Notiffee;
    Celsius temperature() const { return temperature_; }
    . . .
};
```

Sketch the implementation in C++ of a `SensorReactor` class that reacts to sensor changes and turns a `Furnace` on and off. Also, briefly describe how you would deal with the reactor computation taking a significant amount of time. Finally, describe what additional mechanism you would need if the number of sensors was changing over time.

5. **15 points** You join a software company that is just starting to use C++ exceptions and has agreed on the conventions that: 1) every function call that throws an exception must be embedded in a try block. 2) the exception type that is thrown must communicate precisely what went wrong. 3) you must not throw exceptions in response to a call back. Describe the CS249 position on each of these, and why.

The End