

CS 249 MidTerm Exam - Closed Book In-Class Exam

Time allotted: 75 minutes November 1, 2007

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.

The following is a statement of the Stanford University Honor Code:

- A.** *The Honor Code is an undertaking of the students, individually, and collectively:*
- (1)** *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*
 - (2)** *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*
- B.** *The faculty on its part manifests its condence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*
- C.** *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

I acknowledge and accept the Honor Code.

(Signature)

(Print name)

1	2	3	4	5	Total
15	15	15	15	15	75

1. **(15 Points)** Cheriton makes a big deal of the fact that software development is iterative and claims the course techniques help with that. Describe three key techniques presented in the course that aid with iterative development and describe how they do.

2. **(15 Points)** Cheriton's attribute-only approach, in going beyond the basic virtual variable model, calls for further constrained semantics for accessors and mutators, justified in part based on distributed implementation. List three such semantic restrictions and describe how they are justified (if not required) by a distributed implementation.

3. (15 Points) You join a new company and sing the praises of separating processing logic from state, and consequently are handed the the challenge to demonstrate how to do so with following code:

```
class Sensor : public NamedInterface {
    virtual void print( Printer * );
    virtual void onReconfig( Config * );
    . . .
};
class Gps : public Sensor {
    void print( Printer * );
    void onReconfig( Config * );
    ...
};
class Imu : public Sensor {
    void print( Printer * );
    void onReconfig( Config * );
    ...
};
```

That is, the "print" function is called to cause these sensors to print themselves to the designated printer and the "onReconfig" is called when the system configuration is changed, affecting the sensor, with the processing logic for both implemented in these classes. Sketch revised class definitions that would provide this separation and indicate any additional mechanism required to support this revised design. We are not looking for a code-complete answer but just enough to allow a knowledgeable programmer to understand how the revised design would work. You can use pictures, words and code.

Continue for problem 3

4. **(15 Points)** Fearlessly battling conventional wisdom, the course tries to make the case against having a HAL and the call-down/notify-up structure, arguing for the call-up/notify-down model.
- (a) If you had to convince some irritating baby-boomer know-it-all wonder-how-he-got-promoted senior software engineer at your company that call-up-notify-down was the right approach, describe a compelling example you would use.
 - (b) Then, he whines that "up" and "down" don't make sense in general in software. Describe the advantages of this approach in general.
5. **(15 Points)** In the simulation of a swamp full of snails and bacteria, one approach (the Entity approach) is to create an instance of an Entity type for each snail and each bacterium while another (the Value approach) is to just represent the swamp as a matrix (or matrices) of values corresponding to a grid of the swamp area, each value a count of instances of the species in that portion of the swamp (grid).
- (a) Describe how these two approaches can be related according to the relationship between entity types and value types described in the course.
 - (b) Justify the distinction of a named description type from entity type and value type (or else argue that the introduction of this third category of type is not warranted).