

Quiz 2, CSCI 360, Fall 2004

Name: _____

1. [9 pts] Traditional Smalltalk implementations use an inefficient technique for determining where to go when a virtual method is called, a technique that involves a search through several tables at runtime. Explain why they cannot simply use the much faster VMT technique used in most C++ and Java implementations.

2. [12 pts] At right is a class representing a figure on a screen. Suppose we have a thread that sometimes calls `draw` for drawing the figure, and another that calls `shift` when it is time to shift the figure. Describe a specific situation where this code misbehaves, and modify the code to repair the problem.

```
public class Sprite {
    int position;

    void draw(Graphics g) {
        g.setColor(Color.black);
        g.fillOval(position - 10, 90, 20, 20);
        g.setColor(Color.white);
        g.fillOval(position - 5, 95, 10, 10);
    }

    void shift(int delta) {
        position += delta;
    }
}
```

3. [10 pts] Define the term *deadlock* as it relates to thread programming.
4. [9 pts] Give an example of a lambda expression that does not reduce to normal form using applicative evaluation order (also called eager evaluation), but which does reduce to normal form using normal evaluation order.
5. [9 pts] Give a lambda expression representing the *composition* function. That is, the function represented by your expression should take two functions f and g as arguments and return the composition of the two functions — i.e., a function h where $h(x) = f(g(x))$ for any x .