

## Final, CSCI 360, Fall 2004

Name: \_\_\_\_\_

1. [10 pts] For the Java program at right, suppose we have two threads, a and b, where a executes the runA method and b executes the runB method. Give an example of code for runA and runB which could result in deadlock between the two threads.

```
public class Deadlock {
    Thread a;
    Thread b;

    public void runA() {

    }

    public void runB() {

    }
}
```

2. [7 pts] Give a lambda expression representing the function that takes a value  $v$  and returns the *projection function* for  $v$ . That is, given  $v$ , it should return a function that will take any other function  $f$  and return  $f v$ .
3. [6 pts] Define a Scheme function HYP that takes two lengths and returns the length of the hypotenuse of the right triangle whose legs have those lengths. Recall that this length is  $\sqrt{a^2 + b^2}$  for sides  $a$  and  $b$ , and recall that Scheme has a built-in sqrt function to compute the square root of a number.
4. [7 pts] Define the term *purely functional language*. That is, what property must a language have to be called purely functional?

5. [7 pts] Write a Haskell function `replicate` that takes an integer  $n$  and a value  $v$  and returns a list containing  $n$  copies of  $v$ . (This function is actually pre-defined in the Prelude, but of course you shouldn't use the pre-defined function to define this one.) For example:

```
replicate 3 "a" should return ["a", "a", "a"]
```

6. [4 pts] What is the type of the `replicate` function described in the previous question? Use Haskell syntax to describe the type.
7. [10 pts] Write a Haskell function `insert` that takes an integer  $n$  and a list  $\ell$  of integers assumed to be in increasing order, and then returns a list containing the same elements of  $\ell$  but with  $n$  inserted so that the integers are still ascending. For example:

```
insert 4 [2,3,5] should return [2,3,4,5]
```

8. [8 pts] Explain the efficiency loss that occurs with referencing *lazy parameters*.

9. [6 pts] Which parameters for the following function are strict?

```
multAll n data = if data == []  
                 then []  
                 else (n * head data) : (multAll n (tail data))
```

10. [4 pts] Translate the following syntax using Haskell's `do` notation into the equivalent using the Monad class's `>>=` operator.

```
do a <- getLine
    b <- getLine
    putStrLn b
```

11. [7 pts] Define a Haskell function `skipToLine` declared as

```
skipToLine :: Integer -> IO String
```

Given an `Integer`, `skipToLine` should read that many lines from the user using the `getLine` function of type `IO String` and return the `IO String` returned by the final call to `getLine`.

12. [10 pts] Describe the translation process that a compiler can use to eliminate recursive calls from a tail-recursive function.

13. [7 pts] Explain how the reference counting technique works for identifying garbage memory to be deallocated.

14. [7 pts] Explain why copy collection can be a faster technique for garbage reclamation than mark and sweep, even though the copying process is significantly more complex.