CSE 341 Midterm — Answer Key — Spring 2002

This exam is closed book and notes. 64 points total.

1. (10 points) Suppose the following Miranda script has been filed in.

```
plus x y = x+y

append [] ys = ys
append (x:xs) ys = x : append xs ys

my_map f [] = []
my_map f (x:xs) = f x : my_map f xs

tree * ::= EmptyTree | Node * (tree *) (tree *)

treemap f EmptyTree = EmptyTree
treemap f (Node n left right) = Node (f n) (treemap f left) (treemap f right)

treefold f id EmptyTree = id
treefold f id (Node n left right) =
   f n (f (treefold f id left) (treefold f id right))

|| define a few trees

t1 = EmptyTree
t2 = Node 7 EmptyTree (Node 4 EmptyTree EmptyTree)
t3 = Node "aa" (Node "bb" EmptyTree EmptyTree) (Node "cc" EmptyTree EmptyTree)
t4 = Node 10 EmptyTree t4
```

   What is the **type** of the following Miranda expressions? If there is a compile-time type error, say so.

   (a)
   ```
   treemap ::
   (*->**)->tree *->tree **
   ```
   (b)
   ```
   treefold ::
   (*->*->*)->*->tree *->*
   ```
   (c)
   ```
   treefold plus 0 ::
   tree num->num
   ```
   (d)
   ```
   treefold append [] ::
   tree [*]->[*]
   ```
   (e)
   ```
   my_map plus [1..] ::
   [num->num]
   ```

2. (10 points) Given the same script as in Question 1, what is the **result of evaluating** the following Miranda expressions? If there is a compile-time error, or a run-time error, or a non-terminating computation, say so. If the result is infinite, show some of what Miranda would print (enough to see the pattern).

   (a) `treemap (plus 1) t2`
   ```
   Node 8 EmptyTree (Node 5 EmptyTree EmptyTree)
   ```

   (b) `treemap (plus 1) t4`
   ```
   Node 11 EmptyTree (Node 11 EmptyTree (Node 11 EmptyTree (Node 11 ...
   ```

   (c) `treefold plus 0 t2`
   ```
   11
   ```

   (d) `treefold plus 0 t3`
   ```
   type error in expression
   cannot unify tree [char] with tree num
   ```
   (However, you just need to say compile-time error to have your answer counted as correct.)

   (e) `treefold append [] t3`
   ```
   aabbcc
   ```

3. (12 points) Pedagogically valuable true/false questions! (Ken said I wasn't supposed to call them tacky any more.)

   (a) Miranda and applicative order evaluation will always give the same result. False

   (b) Miranda and normal order evaluation will always give the same result. True

   (c) Miranda will always evaluate each subexpression the same number of times as normal order evaluation. False

   (d) Miranda is statically typed. True

   (e) Miranda is type safe. True

   (f) Java is statically typed. True

   (g) Java is type safe. True

   (h) Java requires type declarations for all variables. True

   (i) A programmer can add new methods to class Object in Java. False

   (j) A programmer can define a new class that extends the class Object in Java. True

   (k) The Java runtime stores objects that are bound to global variables in the heap. True

   (l) The Java runtime stores objects that are bound to local variables in a method on the stack. False

4. (10 points) Consider the following recursive function definition in Miranda:

```
squid []     = []
squid (x:xs) = (x+2) : squid xs, if x>0
             = squid xs, otherwise
```

Write an equivalent function using a list comprehension instead of recursion.

```
squid s = [ (x+2) | x <- s ; x > 0 ]
```

5. (10 points) Briefly explain in words the difference between overloading and overriding.

When applied to methods in an object-oriented language, “overloading” means that there are several methods defined in a class or its superclasses with the same name, but that are different methods, disambiguated by the the types of the arguments. Also when applied to methods in an object-oriented language,“overriding” means that there is a method in a subclass that overrides (or supercedes) a method of the same name in the superclass. Thus the method in the superclass is not found when that method is invoked on an instance of subclass (except when using “super”).

Actually, “overloading” applies to non-object-oriented languages as well. In C, for example, the + operator is overloaded – it can be used with integers or floats, and the compiler decides which is to be used based on the types of the arguments. (You don’t need to point this out to have your answer counted as correct, however.)

6. (12 points) Consider the following class in Java.

```
class Counter {

  private int count;

  public Counter()
    {count=0;}

  public void increment(int i)
    {count=count+i;}

  public int contents()
    {return count;}

  public boolean equals (Counter p)
    {return (count==p.contents());}

}
```

What does this print? (It should print two integers and four booleans.)

```
Counter c1 = new Counter();
Counter c2 = c1;
Counter c3 = new Counter();

c1.increment(5);

System.out.println(c1.contents());
System.out.println(c2.contents());
System.out.println(c1==c3);
System.out.println(c1.equals(c3));

c3.increment(5);
System.out.println(c1==c3);
System.out.println(c1.equals(c3));
```

```
5
5
false
false
false
true
```