Homework 1: Recursion

Due Thursday, Feb. 12 at 9:50 am

Keep a copy of your solutions—they'll help you with HW2.

Ground rules for all CS334 assignments:

a. Write your name, the assignment due date, and the number of the assignment at the top of the first page.

b. At the bottom of your assignment, describe any collaboration or assistance that you received, even if it was from a TA or professor.

c. At the bottom of your assignment, write approximately how many hours it took you to complete. I use this to adjust the length of future assignments. The amount of time taken will not affect your grade.

d. If you have any doubt whether your handwriting is legible, type your solutions.

1. Programming

(16 points)
For the following problems, write each solution as a static method. You may check your work using a computer. To compile in the Unix lab, save your program from your favorite editor (mine is Emacs) and then type:

javac *classname*.java
java *classname*

where the specified class contains a public static main method that invokes your actual methods to test them. Do not submit testing code, your main method, or the surrounding class--just hand in the methods required below. If you are sufficiently confident, you can just hand write your solutions clearly and never even put them on a computer. There is no electronic submission in CS334, and you will rarely lose significant points for purely syntactic errors. For the problems on this homework assignment it may help to refer to the online Java documentation, http://java.sun.com/javase/6/docs/.

If you have not used the Unix lab computers before, please see Mary or me to get your password and make sure that you can log in. After logging in, you will be prompted for your choice of window manager. Just hit return to accept the default. (You can experiment with the different choices later.) Pressing the right mouse button will pop up a menu from which you can open a terminal window (xterm), open a web browser (mozilla), and log out.

Before doing anything else, open a terminal window and change your password with the "kpasswd" command. If you feel a little rusty with Unix commands, take a look at http://www.cs.williams.edu/~freund/cs010/assignments/practice1.html

For the following problems, you may not use FOR, WHILE, DO, or any other looping construct. Write static separate methods to:

1. Return the largest of three integers specified as separate arguments named *a*, *b* and *c* (do not use Math.max).

2. Let class IntList be:

```
class IntList {
   public int  first;
   public IntList rest;
   public IntList(int f, IntList r) {
     first = f;
      rest  = r;
   }
}
```

Write code to instantiate a list whose elements are (from first to last): 1 2 3 4 5.

3. Compute the sum of the elements in an IntList named *L*.

4. Compute the length of an IntList.

5. Efficiently compute exponentiation for non-negative exponents by the following recursive algorithm:

$$b^x = \begin{cases} 1 & x = 0 \\ \left(b^{(x/2)}\right)^2 & x \text{ is even} \\ b * \left(b^{(x-1)}\right) & x \text{ is odd} \end{cases}$$

2. Short Answer

(34 points)

6. Give a proof that no compiler can statically determine whether a program prints output, for all programs and inputs. (Hint: use the same structure as the halting problem proof.)

7. Answer the following questions about the Java programming language. A sentence or two is sufficient for each part.

a. Describe two programming errors that the compiler reports.

b. Describe two programming errors that can cause your program to halt with an error message or crash at runtime.

c. What language features do you find most confusing or hard to use?

8. Why did Java's designers decide to make it illegal to name a variable "for" or "if"?

9. java.LinkedList and java.ArrayList implement java.List, and extend Object.

a. What is the difference between **implementing** and **extending** in Java?

Can there ever come a time during program execution when (...and explain why):

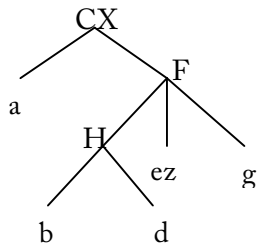
b. a value whose type is LinkedList is bound to a variable declared as a List?
c. a value whose type is ArrayList is bound to a variable declared as a LinkedList?
d. a value whose type is LinkedList is bound to a variable declared as an ArrayList?

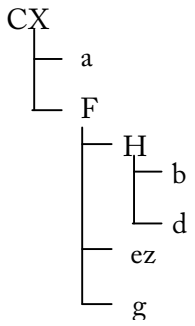
10. What are the advantages and disadvantages of type int not being an Object in Java? What operations occur when the JVM executes the following statement (i.e., what is it semantically equivalent to)?: Integer x = 3;

11. Java contains many types of expressions and statements. For example, FOR loop, method invocation, and class declaration. List as many distinct kinds of Java expressions and statements as you can (note, e.g., that + and – are not separate expressions; they are both binary arithmetic operator application expressions). You can stop if you hit 20 distinct ones. Which ones could you remove and still be able to write programs with the same functionality?

12. What are the advantages of assert being a custom kind of statement in Java (which we call a "special form" in this class), instead of a regular static method on some utility class?

13. A tree can be drawn in many ways, and even expressed in a string using nested expressions. For example, the three trees below are equivalent:






```
"(CX
   a
   (F
      (H
         b
         d)
      ez
      g))"
```

The one on the left looks like something you'd see in a CS theory class (or maybe an upside-down piece of vegetation). The one in the center resembles a file system browser, or a document outline. If you squint, the one on the right looks a bit like a piece of code or HTML. Thinking of those as different applications of the same underlying data structures

The string could also be written more simply as "(C a (F (H b d) e g))". It turns out that converting strings like this into trees is a fundamental operation in implementing a programming language that is called parsing (technically, tokenizing and parsing). Don't actually write the program, but outline the design of a Java program that accepts strings like this and parses them to produces tree data structures. Be sure to list what classes you would to define and what built-in Java classes you would you use.

3. Challenge

(glory, but no points awarded for this)

14. You've definitely used the +, -, and * operators in Java. There are many others that you might not have used. What does the ? operator do in Java? When would you use it?

15. When overriding a method in Java, can you change the argument and return types? If so, what are the constraints? If not, why do you think the language is designed that way?

16. Actually implement part 13.