University of Toronto, St. George Campus
CSC148—Introduction to Computer Science, Fall 1998, Afternoon Section

Second Term Test

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 6 pages (including this one). Answer in the space provided and use the reverse side of the page for rough work; **write legibly**. (If you need more space, use the reverse side of the page and indicate **clearly** which part of your work should be marked.)

Family Name: ______________________________

First Name: ______________________________

Student # : ______________________________

Tutorial: ______________________________

1. ________ / 15

2. ________ / 10

3. ________ / 15

Total: ________ / 40

Question 1

[15 marks in total]

Consider the following `Node` class, which can be used for doubly-linked lists of Strings.

```
class Node {
    public String data;
    public Node prev, next;

    public Node( String initData ) {
        data = initData;
    }
}
```

For this question, you have to write a recursive search method that has the following signature, where `tail` is a reference to the last node of a doubly-linked list.

```
boolean recSearch( Node tail, String tag )
```

(a) (3 marks) Give a precise statement of what it means for the method to be correct, as a function of the input size n (state exactly what n is equal to in terms of the parameters).

(b) (6 marks) Write the body of the method, using recursion. Do **not** use loops!

```
boolean recSearch( Node tail, String tag ) {

}
```

(c) (3 marks) What are the values of `example.prev` and `example.next` after executing this line of code: "`Node example = new Node( "easy" );`"? Why?

(d) (3 marks) If we changed the `Node` class to hold a reference to an `Object` instead of a `String`, would we need to change the body of the method `recSearch` to look for a String in a doubly-linked list? If so, explain what changes should be made; if not, explain why.

Question 2

[10 marks in total]

(a) (4 marks) You're working for a company that needs to design an adventure game. Part of the game involves people, wolves, and werewolves. (A *werewolf* is a creature capable of assuming the form of a person or of a wolf at will.)

Pick the most flexible, general, and maintainable design for these concepts, from among the following choices (check only one). (Note that it **is** possible in Java to have one interface extend another.)

- ☐ We have a general `Creature` class; two subclasses of `Creature`, `Person` and `Wolf`; and a subclass `Werewolf` that extends both `Person` and `Wolf`.

- ☐ We have a general `Creature` interface; two interfaces `PersonInter` and `WolfInter` that extend `Creature`; a class `Person` that implements `PersonInter` and a class `Wolf` that implements `WolfInter`; and a class `Werewolf` that implements both `PersonInter` and `WolfInter`, and has instance variables of type `Person` and `Wolf`.

- ☐ We have a general `Creature` interface; two classes that implement `Creature`, `Person` and `Wolf`; and a class `Werewolf` that implements `Creature`, and has two instance variables of type `Person` and `Wolf`.

- ☐ We have a general `Creature` class; two interfaces, `PersonInter` and `WolfInter`; a class `Person` that implements `PersonInter` and a class `Wolf` that implements `WolfInter`; and a class `Werewolf` that implements both `PersonInter` and `WolfInter`, and has instance variables of type `Person` and `Wolf`.

(b) (4 marks) Explain the main reason why abstract data types should be represented by interfaces instead of classes.

(c) (2 marks) Write your name and student number at the top of every page of this test.

Question 3

[15 marks in total]

Consider the following method (where "`**`" represents exponentiation, so that `c**(a+b)` is equal to `c` raised to the power of `a+b`, for example).

```
// Compute the "discrete binary logarithm" of x > 0, i.e.,
// return an integer y such that 2**y <= x < 2**(y+1).
// -------------------------------------------------------------
public static int discBinLog( int x ) {

    // Pre:  x > 0
    int y = 0;
    int z = 2;
    // Inv:  2**y <= x  &&  z == 2**(y+1)
    while ( z <= x ) {
        y += 1;
        z *= 2;
    }
    // Post:  2**y <= x < 2**(y+1)

    return y;
}
```

(a) (2 marks) If we want to prove that a loop is correct, we must find a bound function `t`; what properties must `t` satisfy? (Just state the properties.)

(b) (4 marks) If we want to prove that a loop is correct, what relationship must we prove between the loop invariant, the loop test, and the loop body? (Just state the relationship, do **not** prove it.)

(c) (3 marks) If we want to prove that a loop is correct, we must show that the loop invariant is true if we assume that the preconditions are true and that we execute the initialization. Prove this for the loop in `discBinLog`.

(d) (4 marks) If we want to prove that a loop is correct, we must show that the postcondition is true if we assume that the loop invariant is true and that the loop test is false. Prove this for the loop in `discBinLog`.

(e) (2 marks) Pick the most appropriate bound function for the loop in `discBinLog` from the following list (check only one).

- ☐ `t = x - z - 1`
- ☐ `t = z - x - 1`
- ☐ `t = x - z`
- ☐ `t = z - x`
- ☐ `t = x - z + 5`
- ☐ `t = z - x + 5`