

CS 2510 Exam 2 – Spring 2012

Name: _____

Student Id (last 4 digits): _____

- Write down the answers in the space provided.
- You may use all syntax that you know from *FunJava* (that is, the parts of Java we have studied in class), although there are several features you will *not* need.
- When defining methods, you do not need to give a complete class definition—just indicate in which class your method definition should be placed.
- For tests you only need to provide the expression that computes the actual value, connecting it with an arrow to the expected value. For example `s.method() -> true` is sufficient.
- Remember that the phrase “design a class” or “design a method” means more than just providing a definition. It means to design them according to the **design recipe**. You are *not* required to provide a method template unless the problem specifically asks for one. However, be prepared to struggle if you choose to skip the template step.
- We will not answer *any* questions during the exam.

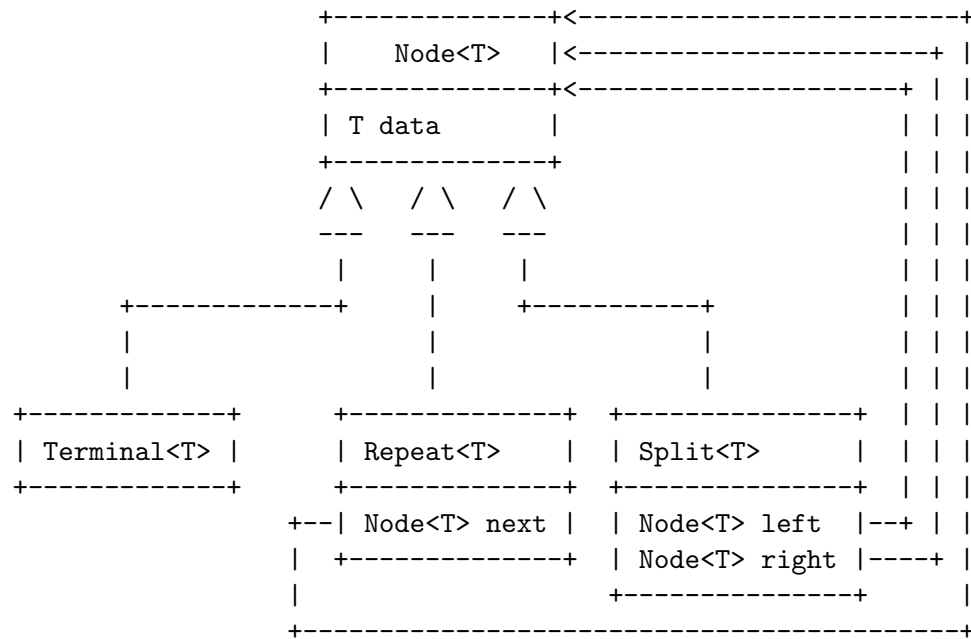
Problem	Points	/
A		/ 5
B		/ 5
C		/10
D		/10
Total		/30

Good luck!

Problem 1

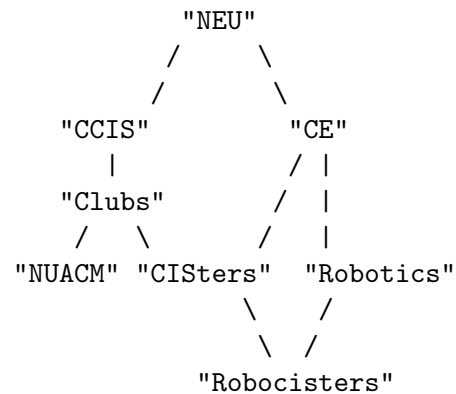
Amateur radio (aka ham radio) broadcasts work as follow: there are three kinds of transmitter nodes: those that simply transmit a signal to another node, those that split a signal and transmit it to two other nodes, and finally those that are the end points of communication that don't broadcast any further. When a broadcast goes out, the network is set up in such a way that no broadcast goes to the same node twice (there are no cycles in the network).

The following class diagram represents a radio broadcast network:



A. (5 points)

Here is a graphical view of a local ham radio network:



When a node appears above another with a line between them, it means the upper node broadcasts to the lower one. The strings given show the data each node carries.

Translate the above information into examples of data that represent the network.

B. (5 points)

Here is the data representation of another network (this one carries integers rather than strings):

```
Node<Integer> h = new Terminal<Integer>(800);
Node<Integer> b = new Terminal<Integer>(12);
Node<Integer> a = new Split<Integer>(1, h, b);
Node<Integer> z = new Terminal<Integer>(7);
Node<Integer> c = new Split<Integer>(9, z, a);
Node<Integer> j = new Repeat<Integer>(11, a);
Node<Integer> d = new Repeat<Integer>(44, j);
Node<Integer> g = new Split<Integer>(19, c, d);
Node<Integer> t = new Terminal<Integer>(4);
Node<Integer> s = new Split<Integer>(13, t, c);
Node<Integer> k = new Split<Integer>(14, s, g);
```

Translate the above into a graphical representation of the network similar to that given in part A.

C. (*10 points*)

One measure of the networks robustness is the number of paths from a node to any terminal.

Design the method `countAllPaths` that computes the number of paths from this node to a terminal node.

D. (*10 points*)

The quality of radio transmissions deteriorate as they are broadcast across more and more nodes. We would like to measure the degree of distortion in a network.

Design the method `distanceTo` that finds the length of the shortest network route from this node to the node containing the given data.