Assignment 1

1. Familiarize yourself with the NetBeans compiler using online and web resources. On the classroom computers, you should be able to launch NetBeans using the command `netbeans` in a Terminal window. If this does not work, try `/usr/local/netbeans/bin/runide.sh`.

2. Write a tree class in Java using NetBeans. The class should have a constructor that takes a string containing a Newick-format tree description and converts it into one of the tree data models covered in the lecture (the binary, polytomous or Felsenstein tree models; we recommend the binary model). Minimally, the constructor should handle rooted and unrooted binary trees with branch lengths (you can use the basal trichotomy convention described in the lecture notes to determine whether the tree is rooted or unrooted before actually constructing and connecting the nodes). The tree class should also have a Print function that takes an output stream (a PrintWriter object) as an argument and writes a Newick-format tree to that output stream. Assume that unrooted binary trees are written as if they were rooted on one terminal and using a basal trichotomy, that is, a four-tip tree grouping A and B on one side and C and D on the other would be written as ((A,B),C,D) assuming that it was rooted on D. This is the format used by most existing programs including e.g. Felsenstein's retree and MrBayes.

3. Write a wrapper class that asks for a file name and then tries to read this file either as containing a list of Newick-format trees, one on each line, or as a Nexus-formatted file containing a trees block. The wrapper class should create a new tree for each tree description it encounters in the tree file, and these trees should be stored in memory. The wrapper class should then ask the user for a second file name, and write a Nexus-format trees block or a simple list of Newick-format trees to that file. The list of trees should contain all the trees read from the input file.

Make sure that your code is well documented and follows good programming practices. Mail the source code and a jar file with your program to Peter or Fredrik no later than **Monday September 12**.