

## Homework 15

See the code provided in the HW15Assigned/userid directory, which implements a Node class that allows trees to be built. Convert this code into a Generic class that is parameterized by a single type E that is the type of the *data* field in Node. Types for E should be restricted to types that extend from Comparable.

You should create a generic class TwoTuple that extends java.lang.Comparable, and that holds two values, i1 and i2. i1, i2 and TwoTuple should all extend Comparable. Given TwoTuple objects o1 and o2,

**o1 < o2** if o1.i1 < o2.i1 or o1.i1 == o2.i1 and o1.i2 < o2.i2.

and

**o1 == o2** if o1.i1 == o2.i1 and o1.i2 == o2.i2

otherwise o1 > o2.

Your Node should be able to hold TwoTuple objects, and form a tree of TwoTuple objects.

I would strongly suggest that when doing your homework, you comment out the lines in HW15.java that use TwoTuple, get the program working with the Node generic, and then implement the TwoTuple generic.

### What to turn in:

Turn in your code in a directory called <userid>, where <userid> is your Purdue login/userid. javac HW15.java executed in the **userid** directory should allow this code to compile, and java HW15 executed in the **userid** directory should execute your program. Zip up the userid directory and turn it in.

### Points for grading:

3 points for creating a working TwoTuple.

3 points for creating the Node class

2 points for the correct output. No extra points should be taken off if the only output difference is from not having created your own class that extends Comparable.

1 point for S1 giving an error similar to the one in S1err.txt, when uncommented.

1 point for S2 giving an error similar to the one in S2err.txt, when uncommented.