

## Homework 10

See the code provided in the HW10CPPAssigned directory, which implements a Node class that allows trees to be built. Convert this code into a templated class that is parameterized by a single type T that is the type of the *data* field in Node. You can assume that any type used to instantiate a Node has the <, >, == and << operators defined.

You should create a templated class TwoTuple that holds two values, i1 and i2. i1, i2 and TwoTuple should all have operators ==, < and > defined on them. TwoTuple should also define operator<<, which is given to you. Note that its definition is inlined into the .h part of the template definition – this is the preferred way to do this. 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.i2 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 HW12.cpp that use TwoTuple, get the program working with the Node template, and then implement the TwoTuple template.

You are also provided Int.h Int.cpp and main.cpp. You should need to change these files except to comment out things in main.cpp, as suggested in the last paragraph.

*Remember to always define the .h and .cpp parts of a templated class C in the file C.h.*

### What to turn in:

Turn in your code in a directory called <userid>, where <userid> is your Purdue login/userid. g++ \*.cpp executed in the **userid** directory should allow this code to compile, and running ./a.out should execute your program. Zip up the userid directory and turn it in.

### Points for grading:

4 points for creating a working TwoTuple.

4 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 the TwoTuple class – the four points lost above will be sufficient.