ETEC2101	Lab6: **Graphs**	Points: **100** (up to 130)
Assigned: **10/31/2016**	Due: **11/16/2016 at noon (no late work)**	

Tasks:

1. (**35 points**) Create a **Graph** class in the **etec2101** package.
   a. This class should be generic in two ways: **N**, the node-data and **E**, the edge-data.
   b. You'll likely create a Node and Edge class that have a value of type N and E, respectively (along with any other structural attributes)
   c. There shouldn't be any drawing code here!
   d. You can decide what internal graph representation you want to use (adjacency list, adjacency matrix, etc.)
   e. Create at least the following public methods:
      i. addNode(N val)
      ii. addEdge(E data, N nodeA, N nodeB, boolean bidirectional);
2. (**35 points**) Create a new class, **DrawableGraph** that extends from Graph and is in a Lab06_xx package.
   a. Create two classes: **DrawableNode** and **DrawableEdge**
      i. Both should have some kind of draw method.
      ii. Your DrawableGraph will use these as the N and E "parameters" to extends.
   b. This method should be capable of loading a map similar to those on the web site.
   c. This class should include some kind of draw method which draws all nodes and edges
3. (**10 points**) Create a main program class (also in Lab06_xx package)
   a. You'll likely copy / paste one of your slick solutions from a previous lab (or the web page)
   b. Create a DrawableGraph instance and draw it!
4. (**10 points**) Graphical effects:
   a. Scale the map to fit the screen
   b. Draw arrow-heads on the edges to indicate direction visually (maybe dust off some 1803 ideas☺)
5. (**20 points**) Implement either Depth-first or Breadth-First Traversal
   a. Draw the result visually.
   b. Use the parentTree method we discussed in class.
   c. The method to accomplish this should be in the Graph class, but the drawing should be done in DrawableGraph
   d. When doing traversals, shade by depth (i.e. farther away are a bit darker)
6. (**10 points**) Implement the other Traversal method too.
7. (**10 points**) For good design and documentation / commenting (I'll only look for JavaDocs in the Graph class)
8. Ask if you'd like to see a demo…