CPSC 220 11/18

Using Java's tree class

```
import java.util.TreeMap;

public class Temp {
	public static void main(String[] args) {
		TreeMap<String, Integer> treeMap =
			new TreeMap<String, Integer>();
		treeMap.put("One", new Integer(1));
		treeMap.put("Two", new Integer(2));
		treeMap.put("Three", new Integer(3));
		System.out.println(treeMap.get("Two"));
	}
}
```

a map uses keys and values

must specify type of key and value

use put method to add

use get method with a key to get a value

Adjacency List Graph Implementation

```
import java.util.ArrayList;

public class Graph<T> {
    ArrayList<Node<T>> nodes = new ArrayList<Node<T>>();

    public void addNode(T object) {
        nodes.add(new Node<T>(object));
    }

    public void addLink(int fromIndex, int toIndex) {
        addLink(nodes.get(fromIndex), nodes.get(toIndex));
    }

    public void addLink(T fromObject, T toObject) {
        addLink(find(fromObject), find(toObject));
    }

    public void addLink(Node<T> fromNode, Node<T> toNode) {
        if(fromNode != null && toNode != null) {
            if(!fromNode.links.contains(toNode))
                fromNode.links.add(toNode);
        }
    }

    private Node<T> find(T object) {
        Node<T> foundNode = null;
        for(Node<T> node : nodes) {
            if(object.equals(node.object))
                foundNode = node;
        }
        return foundNode;
    }

    class Node<T> {
        protected T object;
        protected ArrayList<Node<T>> links;

        public Node(T object) {
            this.object = object;
            this.links = new ArrayList<Node<T>>();
        }
    }

}
```