BST Code Testing Approach:

NOTE: THIS IS DUE TUESDAY, APRIL 4
(It is not due our normal Friday because you must do this BEFORE you work on you BST project.

While it is due Tuesday, it may be beneficial to finish writing this earlier than that so you can start working on the BST project.)

This is the first part of the next project.

In previous labs I have written the test cases for you. I explicitly told you to write specific methods first, then test them with a specific set of data, and then check to make sure your code was working by comparing your output with my output.

In this project, you will be writing your own test cases.

Notes:

- At the end of this file, you can see the content of the file you will use for testing.
    - In past labs, I had a method that read in the data from a testing file. You may assume I will give you a method that will read in the file.
- Each animal in the file has 3 fields: name, status (level of threat), and some info about the animal.
    - You will be creating a binary search tree based on the name.
- You may assume that the readfile method I will give you reads in all the information from the test file and calls the insert method you will be writing. Insert inserts each animal into a binary search tree.
    - Thus you have an empty tree until you insert all the data from the file.
- UpdateStatus Method – this JUST changes the data in a node. It doesn't move, remove, insert. It just changes an animals status. For instance, an animal might be Critically Threatened T1, but because we're being really good about restoring its habitat, it's status changes to just Threatened T2. That's what you ae doing in Update Status.
- And SetHeights Method. This updates the heights of the ancestor nodes when you add a new node to the tree. We are not using the height of a node in this lab. However, it is critical for balancing a tree. Even though we are not implementing the balancing of a binary search tree, it is important that you be aware of how to set the heights for when you do want to balance a binary search tree.

Now Begin:

Your next project involves writing a binary search tree for endangered species. You'll be adding to, removing from, finding in, and updating nodes in the endangered species tree.

Methods you will be writing:

- **BST** (The constructor)
- **setHeight**(sets the height of each node)
- **Find** (finding data in a tree)
- **printTreeIO** (printing the tree using in-order traversal)
- **printTreePre**(printing the tree using pre-order traversal)
- **printTreePost**(printing the tree using post-order traversal)
- updateStatus(updates the data at a node in the tree)
- **remove** (removes a node from the tree – used for removing a node with 2 kids and/or calling removeNoKids or removeOneKid (whatever is appropriate))
- **removeNoKids**(removes a node with no kids)
- **removeOneKid**(removes a node with only one kid
- **Insert (to insert new data (aka a new node) in the tree)**

In the labs so far, I have given you clear instructions on which method(s) to write first, and, after you've written them, what the test output should look like so you could know whether your methods are working properly.

This time you will be doing that part.

Part 1)

Determine the order in which you intend to write the methods. There is more than one order that will work just fine, but, for instance, you might want to write things like the constructor, the insert, and one of the printTree methods early on so you can successfully create a tree and test that the tree was created successfully before you do anything else.

Fill in the blanks, below, with the order in which you intend to write the above methods:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____

Part 2:

Now create your own test output – at what points do you intend to test your methods, and what output are you expecting to get. You will be using as input the test data I gave you at the end of this document.

A couple of notes:

1) Some of the methods may not be testable until you write another method. In other words, you probably can't know whether you wrote the constructor and the insert method correctly until you write at least one of the printtree methods so you can see what your tree looks like.
2) The data type in each Node is Species, and consists of 3 fields:
   - The animal's name,
   - its status, and
   - info about the animal.

   All three fields are strings

   You will be inserting/finding based on the animal's **name**.

   Again, strings can be compared using > and < and =, just like numbers.

   > apple < bear < petunia < to < zebra

   (It's alphabetical order, with strings that come first alphabetically being less than the strings coming after them

Here is the data you will be using for your test cases. The | indicates a separation of the animal's name, its status, and the info about the status (I used this for reading in the data from a file – you don't have to worry about it at all!)

AND: FYI only:

- EX = Extinct
- EW = Extinct in the wild
- T1 = Critically Endangered
- T2 = Endangered
- VU Vulnerable
- LT = Least Threatened

```
/***************************************************************************/
/* The test data file                                                      */
/***************************************************************************/
```

Tiger |T1| The tigers are critically threatened. It's not good.|

Orangutan |T2| Orangutans are also in trouble. Because of the whole habitat loss thing.|

Giant Panda |VU| Yep, Giant Pandas are threatened too, although less so now. They're very cute.|

Blue Whale |T2| Blue Whales are in trouble - they eat a lot, they live in the sea, seas are kind of a mess right now.|

Asian Elephant |T1| These are the elephants with the tiny ears. Everyone wants their tusks. Very sad.|

Gorilla |VU| They were on the list of endangered species, so I have to believe it is true. Jane Goodall is not happy.|

Snow Leopard |T1| Makes sense that these are endangered. They have snow in their name, and there's the whole global warming thing.|

Rhinoceros |T1| These are also endangered because of their tusks. What is it about tusks?? Just let the animal that grew 'em keep 'em!|

Sea Turtle |VU| These animals have been around for 100 million years, and, yep, we're about to wipe them out. Quite an accomplishment, in a bad way...|

Sea Otter |T1| They live near Alaska mostly. There used to be 300,000. Now? 3000.|

African Wild Dog |T1| Okay, so this particular dog isn't terribly cute, but it is really seriously endangered, and I don't wish for it to be extinct!|

Amur Leopard |T1| This animal is seriously hurting because it has a beautiful coat, and people keep hunting it for its coat. Like the leopard doesn't need its coat. Let the leopard keep its coat, people!|

Axolotl |T2| No idea what this is. But it was on the list of endangered species. Should probably google it.|

Black-footed Ferrets |T2| Get this - this ferret is only found in North America and eats animals its own size. |

Chimpanzees |VU| Yep, we're wiping out the habitat of the species most closely related to us. Good job humans!|

Galapagos Penguins |T2| My guess is that the Galapagos Islands are going to be under water shortly, so any animal with the word 'Galapagos' in their name is probably in trouble.|

Golden Lion Tamarins |VU| Poor Golden Lion Tamarins. Their habitat is going away. Poor, poor Golden Lion Tamarins.|