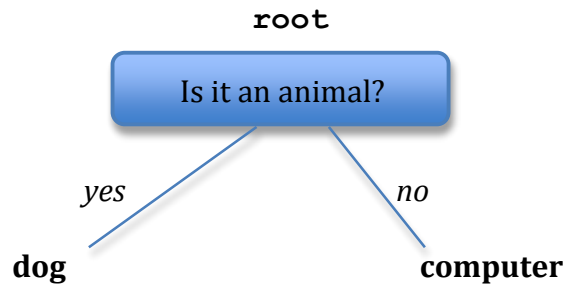


Twenty Questions with Trees

- (0) Write down a **draft** of your algorithm for your `main()`, that is, the **algorithm** which describes how you will “keep playing” the game Twenty Questions. You may work with a partner to discuss how to *design* your program. Note: don’t get hung up on how you will implement the nodes and tree; rather, focus on the user interaction that is needed in order to keep playing a game. To get started, you can assume you have part of your tree already built as shown below:



- (1) In our link list implementations so far, we have used a `struct` for the NODEs. We *really* should use a class for the Nodes (rather than a **struct**). **Why?**
- see the `Node.h` and `Node.cpp` files -- study them!
(Note that the `PRE/POST` are defined in the `Node.h` file)
 - You’ll need to implement the functions in `Node.cpp` that are incomplete
For now, do not concern yourself with your `Tree` class. Work on your Nodes.
- (2) Write a `testNode.cpp` (main) file that will test each of the methods in the `Node` class, one at a time; for example: “now testing COPY CTOR:”.
- (3) Open the `Tree.h` and `Tree.cpp` files -- study them!
- (4) Write a `testTree.cpp` (another, different main) file which will test each of the methods in the `Tree` class one at a time.
- (5) At this point, you should be able to use `Tree` and `Node` classes to write your `main()` to play Twenty Questions.

“I’m thinking of an object”