

CSE 143 Java

Footnote To Trees: Inner Classes

3/14/2003

(c) 1997-2003 University of Washington

15-1

A Programming Dilemma

- The nodes we've defined so far for linked lists and trees have been public classes with public instance variables:

```
public class BTreeNode {  
    public Object item;           // data item in this node  
    public BTreeNode left;       // left subtree, or null if none  
    public BTreeNode right;      // right subtree, or null if none  
    public BTreeNode(Object item, BTreeNode left, BTreeNode right) { ... }  
}
```

- This simplifies examples, and increases performance... but it's very bad practice.
- When one class (like a node) is used only as a helper to another class..
 - It would be ideal to keep it inaccessible to the outside, without giving up programming convenience or speed.

3/14/2003

(c) 1997-2003 University of Washington

15-2

Solution: Inner Classes

- One class may be defined fully within another class
- Called an "inner class"

```
class OuterClass {  
    //constructors, variables, methods... and:  
    class InnerClass {  
        //constructors, variables, methods of InnerClass  
        ...  
    } //end class Inner  
} //end class Outer
```

- Inner class can be marked public, protected, or private
 - Just like instance variables and methods
 - Containing class can always reference its own private instance variables, methods – and inner classes!

3/14/2003

(c) 1997-2003 University of Washington

15-3

Solving the Tree/Node Problem

- Make Node a private inner class of BinTree:

```
public class BinTree {  
    //constructors, variables, methods... and:  
    private class BTreeNode {  
        item;           // data item in this node  
        BTreeNode left;  // left subtree, or null if none  
        BTreeNode right; // right subtree, or null if none  
        BTreeNode(Object item, BTreeNode left, BTreeNode right) { ... }  
    } //end class BTreeNode  
} //end class BinTree
```

- BinTree has full access to the members of BTreeNode
 - Regardless of member public/protected/private marking

3/14/2003

(c) 1997-2003 University of Washington

15-4

More About Java Inner Classes

- We've been using inner classes occasionally without calling attention to it.
 - Point2D.Double means: the (public) inner class named Double of the class named Point2D.
 - Sample code has had examples: TextListener, CountyInformationUnit, etc.
- The inner/outer relationship is not the same as inheritance or composition
 - I.e., neither is-a or has-a
- Inner classes have many interesting twists and turns
 - Inner classes can even be anonymous (unnamed), like objects
 - Topics for another course!