SOLUTION NOTES

Foundations of Programming in Java 2003 Paper 11 Question 3 (FHK)

```
              Java TreeSort Question

Part a:

Within any instantiation of a class, Java permits use of 'this' as a
reference to that instantiation.  Overwriting this reference clearly
cannot be permitted and Java automatically associates the attribute
final to this.  Accordingly, the highlighted assignment statement
provokes a compiler error.

In method main() the programmer sets up a Node tree which has a null
reference and then proposes to use tree.put(8) to assign a value to
this first Node.  Since tree is null, tree.put(8) does not exists so
the assignment statement which offends the compiler would not help even
if it were somehow permitted.

To fix the program it is necessary to set up a proper Node initially
and build up from that.  The put() method, suitably modified, can then
be used for adding further notes.


Part b:

Here is a working version of the program with changes made to method
main() and method put() as well as incorporating the required toString()
method:
```

```
    public class TreeSort
     { public static void main(String[] args)
        { Node tree = new Node(8);
          tree.put(16);   tree.put(4);
          System.out.println("Sorted values:  " + tree);
        }
     }
```

```
class Node
 { private int val;
   private Node left, right;

   public Node(int n)
    { this.val = n;
      this.left = null;
      this.right = null;
    }

   public void put(int k)
    { if (k<this.val)
         if (this.left == null)
            this.left = new Node(k);
         else
            this.left.put(k);
      else
         if (this.right == null)
            this.right = new Node(k);
         else
            this.right.put(k);
    }

   public String toString()
    { String s = ((this.left != null) ? this.left.toString() : "") +
                 this.val + " " +
                 ((this.right != null) ? this.right.toString() : "");
      return s;
    }
 }
```

Part c:

Here is an appropriate method sum() to be incorporated into class Node:

```
   public int sum()
    { int s = ((this.left != null) ? this.left.sum() : 0) +
                 this.val +
                 ((this.right != null) ? this.right.sum() : 0);
      return s;
```

```
        }
```

To use this, an extra statement should be added to method main():

```
          System.out.println("Summed values:  " + tree.sum());
```