

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4
5  // Step 4: Private and public postOrder() methods.
6  template<class Comparable>
7  void BinarySearchTree<Comparable>::postOrder() const {
8      if(isEmpty()) return;
9      postOrder(root);
10 }
11
12 template<class Comparable>
13 void BinarySearchTree<Comparable>::postOrder(BinaryNode <Comparable> *t)
14 • const {
15     if(t == NULL) return;
16     postOrder(t -> left);
17     postOrder(t -> right);
18     cout << t -> element << " ";
19 }
20
21 // Step 6: Private and public height() methods.
22 template <class Comparable>
23 int BinarySearchTree<Comparable>::height( ) const {
24     if(isEmpty()) return -1;
25     height(root);
26 }
27
28 template <class Comparable>
29 int BinarySearchTree<Comparable>::height(BinaryNode<Comparable> *t) const {
30     if(t == NULL) return -1;
31     int heightLeft = height(t -> left);
32     int heightRight = height(t -> right);
33     if(heightLeft > heightRight) return heightLeft + 1;
34     else return heightRight + 1;
35 }
36
37 // Step 8: Private and public numLeaves() methods.
38 template<class Comparable>
39 int BinarySearchTree<Comparable>::numLeaves() const {
40     if(isEmpty()) return 0;
41     else numLeaves(root);
42 }
43
44 template <class Comparable>
45 int BinarySearchTree<Comparable>::numLeaves (BinaryNode<Comparable> *t)
46 • const {
47     if(t == NULL) return 0;
48     else return (1 + numLeaves(t -> left) + numLeaves(t -> right));
49 }
50

```

```

50
51
52 // Step 9 part 1: Private and public isBalanced methods.
53 template<class Comparable>
54 bool BinarySearchTree<Comparable>::isBalanced() const {
55     if(isEmpty()) return 1;
56     else isBalanced(root);
57 }
58
59 template<class Comparable>
60 bool BinarySearchTree<Comparable>::isBalanced(BinaryNode <Comparable> *t)
61 • const {
62     if(t == NULL) return 1;
63     int heightLeft = height(t -> left);
64     int heightRight = height(t -> right);
65     if(abs(heightLeft - heightRight) <= 1 && isBalanced(t -> left) &&
66     • isBalanced(t -> right))
67         return 1;
68     return 0;
69 }
70
71 // Methods included with the assignment have been removed for printing.
72 // They are in Cobra.
73

```