mytree.cpp

```cpp
#include <iostream>
#include "bst.h"
using namespace std;

int main() {

  // Prelude
  cout << "\n" << "\tJulio R. Corzo\n" << "\tCSC245 - Lab 4\n" << endl;

  // Step 1: Instationation of BST with 0 as ITEM_NOT_FOUND.
  BinarySearchTree<int> t(0);

  // Step 2: Populating the tree with the indicated nodes.
  t.insert(6);
  t.insert(8);
  t.insert(2);
  t.insert(1);
  t.insert(4);
  t.insert(3);

  // Step 5: Sample call to postOrder().
  cout << "\tpostOrder() method test\n" << "\tTree t in postOrder: \t";
  t.postOrder();
  cout << endl;

  // Step 7: Checking height() method on tree t and empty tree.
  cout << "\n\theight() method test" << endl;
  if(t.isEmpty()) {
    cout << "\tError: Tree t is empty." << endl;
  } else {
    cout << "\tTree t height: \t" << t.height() << endl;
  }

  BinarySearchTree<int> t2(0);
  if(t2.isEmpty()) {
    cout << "\tError: Tree t2 is empty." << endl;
  } else {
    cout << "\tTree t2 height: \t" << t2.height() << endl;
  }

  // Step 9 part 2: Calls to check to see if isBalanced() is working
  properly.
  cout << "\n\tisBalanced() method test" << endl;
  switch(t.isBalanced()) {
    case 1: cout << "\tTree t is Balanced!" << endl; break;
    default: cout << "\tTree t is not Balanced!" << endl;
  }

  switch(t2.isBalanced()) {
    case 1: cout << "\tTree t2 is Balanced!" << endl; break;
    default: cout << "\tTree t2 is not Balanced!" << endl;
  }
```

```
51     }
52
53     cout << endl;
54   }
55
```