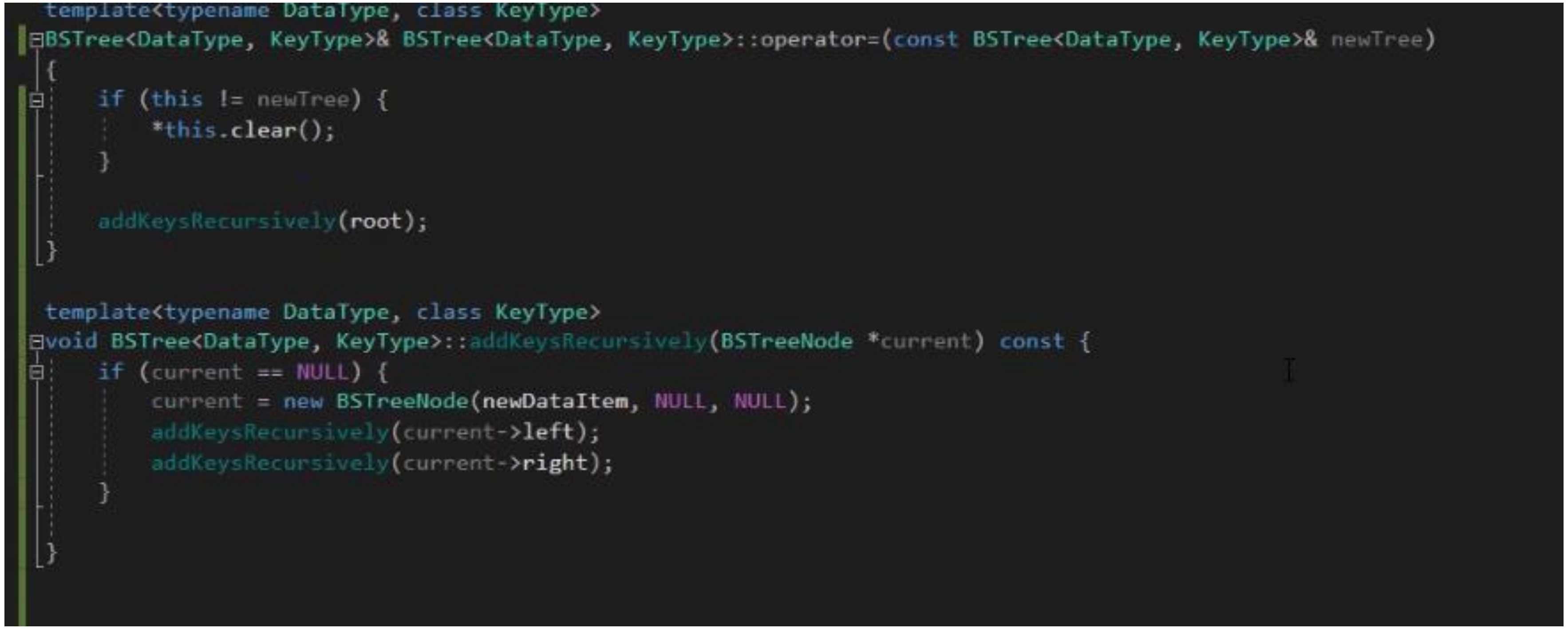
HI

!

This is an informal presentation

Iteration

- For this please assume that the function addKeysRecursivly takes 2 inputs
    - BSTreeNode *current
    - SomeDataType newRoot.
- As you do here you perform the operation before going into a recursive call.
    - Personally I would use a separate null check for each addKeysRecursively call.
- The pattern will be left bottom to right bottom so you can use that to determine where it should be or use the auto sort thing with insertion to determine that.



```
template<typename DataType, class KeyType>
BSTree<DataType, KeyType>& BSTree<DataType, KeyType>::operator=(const BSTree<DataType, KeyType>& newTree)
{
    if (this != newTree) {
        *this.clear();
    }

    addKeysRecursively(root);
}

template<typename DataType, class KeyType>
void BSTree<DataType, KeyType>::addKeysRecursively(BSTreeNode *current) const {
    if (current == NULL) {
        current = new BSTreeNode(newDataItem, NULL, NULL);
        addKeysRecursively(current->left);
        addKeysRecursively(current->right);
    }


}
```