```
#include<iostream>
using namespace std;

class Node {

	Node(int d)
	{
		data = d;
		left = nullptr;
		right = nullptr;
	}
	friend class BinTree;

private:
	int data;
	Node* left;
	Node* right;
};

class BinTree {
private:
	void insert(Node*&, int);
	void inorder(Node*);

protected:
	Node * root;
public:
	BinTree();
	void insert(int);
	void inorder();
};

BinTree::BinTree()
{
	root = nullptr;
}

void BinTree::insert(int val) { insert(root, val); }

void BinTree::insert(Node * &ptr, int val)
{
	if (ptr == nullptr)
		ptr = new Node(val);
	else
	{
		if (val < ptr->data)
			insert(ptr->left, val);
		else
			insert(ptr->right, val);
	}
}

void BinTree::inorder() { inorder(root); }

void BinTree::inorder(Node * ptr)
{
	if (ptr != nullptr)
	{
			inorder(ptr->left);
			cout << ptr->data << " ";
			inorder(ptr->right);
		}
}


int main()
{
		BinTree bt;
		int count = 0;
		bt.insert(7);
		bt.insert(9);
		bt.insert(5);
		bt.insert(6);
		bt.insert(2);
		bt.insert(12);
		bt.insert(8);

		cout << " Inorder = ";
		bt.inorder();

		getchar();
		return 0;
}
```