```
//HW5 Due: 11:59pm, Wednesday, Nov. 8
/*

For the class triangle, implement construcotr, copy constructor,
move constructor, copy assignment,
move assignment, destructor, FirtThreeRow.
Also, implement operator<< to allow cout of triangle object.
See the sample screehshot for output requirement

Only smart pointers are allowed.

No raw pointers allowed.
No extra helper functions allowed.
No external structures allowed.

*/

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

class node {
public:
        int value;
        shared_ptr<node> Lchild;
        shared_ptr<node> Rchild;
        weak_ptr<node> next;
        node(int i) : value(i) {}
        node() {  }

};

class triangle {
public:
        shared_ptr<node> top;
        triangle(int n, int m);//constructor, n levels with node being random values in 0
... m-1
        triangle() { cout << "Default Constructor\n\n"; }
        triangle(const triangle& t);//copy constructor
        void operator=(const triangle& t); //copy assignment
        ~triangle();//destructor
        triangle(triangle&& t);//move constructor
        void operator=(triangle&& t);//move assignment
        triangle FirstThreeRow();//return a triagle which is the first three rows of *this.
        //Assume the triangle *this have three or more than three rows.
};

int main() {

        triangle T1(7, 10);
        cout << T1 << "\n\n";

        triangle T2{ T1 };
        cout << T2 << "\n\n";
        triangle T3;
        T3 = T2;
        cout << T3 << "\n\n";

        T3 = T2.FirstThreeRow();
        cout << T3 << "\n\n";


        return 0;
}
```