Random Generator Sample Program

```
/*
Author: Julius Dichter

Simple Linear Feedback Generator Class
includes a random method for a k-bit random number
and a Chi Square Test method to test quality of the distribution
*/


#include <ctime>
#include <cstdlib>
#include <iostream>

using namespace std;

class Generator
{

public:

        Generator()
        {
                sequence = time(NULL);
                cout << "sequence = " << hex << sequence << dec << endl;
        }

        unsigned long int random(int bitCount)
        {
                int returnValue = 0;

                for (int i = 0; i < bitCount; i++)
                {
                // detect the tapping bits, bit0 and bit3
                // bit0 = 1 only if bit 0 in the sequence is a 1
                // bit3 = 1 only if bit 3 in the sequence is a 1

                int bit0 = (sequence & 0x00000001) != 0 ? 1 : 0;
                int bit3 = (sequence & 0x00000008) != 0 ? 1 : 0;

                // determine the value of the XOR

                unsigned long int xorBit = ((bit0 != bit3) ? 0x80000000 : 0);

                // shift the sequence to the left by one bit and OR new bit on the left

                sequence >>= 1;
                sequence |= xorBit;

                // shift the returnValue to the left by one bit and OR new bit on the right

                returnValue <<= 1;
                if (xorBit != 0)
                        returnValue |= 0x00000001;

                }

                return returnValue;
        }
        bool chiSquare(int a [], int N, int R)
        {
                double top = 0;
                double nr = static_cast<double>(N) / R;

                for (int i = 0; i < R; i++)
                {
                        top += pow(a[i] - nr, 2);
                }

                double chi = top / nr;

                double loVal = R - 2 * sqrt(R);
                double hiVal = R + 2 * sqrt(R);

                cout << "Chi Square = " << chi << endl;
                cout << "loVal = " << loVal << "  hiVal = " << hiVal << endl;



                return chi >= loVal && chi <= hiVal;
        }

private:

        unsigned long int sequence;

};
```

Random Generator Sample Program

```
/*
Author: Julius Dichter

Random Number Generator Driver Program
creating 10,000 random numbers in the RANDE of 0-15
*/

#include "stdafx.h"
#include "Generator.h"


int _tmain(int argc, _TCHAR* argv[])
{

       const int BITS = 4;
       const int RANGE = 16;
       const int TOTAL_COUNT = 10000;

       Generator generator;
       int counters[RANGE] = { 0 };

       for (int i = 0; i < TOTAL_COUNT; i++)
       {
              int result = generator.random(BITS);
              //cout << result << endl;
              counters[result]++;
       }


       for (int i = 0; i < RANGE; i++)
              cout << "[" << i << "] = " << counters[i] << endl;

       bool chiResult = generator.chiSquare(counters, TOTAL_COUNT, RANGE);

       cout << "Chi Square Test: " << (chiResult ? "Passed" : "Failed") <<  endl;


       return 0;
}
```