

Forum:[Hack Forums](#)

Thread: [Random String Generator \(Using Boost.Random\)](#)

Trisomy 21 - April 24, 2014, 10:19 pm

I have only just started learning the boost libraries, and I started off by creating a program that will generate any number of strings with any length with random customizable characters. It took me longer than it probably should have to understand some of the uses of boost.Random, hopefully this may be able to help somebody out. Open the spoiler to see the program in action, and the source is below.

Spoiler ([Click to View](#))

```
Code #include <iostream>
#include <fstream>
#include <vector>
#include "boost/random.hpp"
#include "boost/generator_iterator.hpp"

std::vector<int> getRandom(int n, int l, int floor, int ceiling)
{
    std::vector<int> numbers;

    typedef boost::mt19937 RNGtype; //defining as RNGtype to clean up code below
    RNGtype RNG(time(0)); //setting seed

    boost::uniform_int<> range (floor, ceiling); //range of possible numbers
    boost::variate_generator< RNGtype, boost::uniform_int<> > gen(RNG, range); // 'gen' generates
the random number using my defined RNGtype

    for (int i = 0; i < (n * l); i++)
    {
        numbers.push_back(gen()); //storing random numbers in a vector that will be returned to main
    }

    return numbers;
}

void writeToFile(int n, int l, std::vector<char> randomChars)
{
    std::ofstream ofile;
    ofile.open("output.txt");

    for (int i = 0; i < (n * l); i++)
    {
        if (i % l == 0 && i != 0) ofile << "\n";
        ofile << randomChars[i];
    }
    ofile.close();
}
```

```

int main()
{
    const char chars [62] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
'w', 'x', 'y', 'z',
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z',
    '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

    int n, l, floor, ceiling; // 'n' = number of strings, 'l' = length of each string

    std::cout << "How many strings do you want to generate?\n";
    std::cin >> n;

    std::cout << "\nHow many characters in each string?\n";
    std::cin >> l;

    std::cout << "\nWhat characters to include?\n"
        << "[01] All (A-Z, a-z, 0-9)\n"
        << "[02] Uppercase and Lowercase (A-Z, a-z)\n"
        << "[03] Uppercase only (A-Z)\n"
        << "[04] Lowercase only (a-z)\n"
        << "[05] Numbers only (0-9)\n";

    int * choice = new int();
    std::cin >> *choice;
    switch (*choice)
    {
        case 1: floor = 0; ceiling = 61;
            break;
        case 2: floor = 0; ceiling = 51;
            break;
        case 3: floor = 26; ceiling = 51;
            break;
        case 4: floor = 0; ceiling = 25;
            break;
        case 5: floor = 52; ceiling = 61;
            break;
        default: floor = -1;
    }
    delete choice;

    if (floor != -1)
    {
        std::vector<int> randomNumbers = getRandom(n, l, floor, ceiling); //retrieves random numbers
and stores them in 'randomNumbers'
        std::vector<char> randomChars;

        for (int i = 0; i < (n * l); i++) //uses values from 'randomNumbers' and 'chars' to generate/store
random characters in 'randomChars'
        {
            randomChars.push_back(chars[randomNumbers[i]]);
        }
    }
}

```

```

for (int i = 0; i < (n * l); i++) //prints the values stored in 'randomChars' to the console
{
    if (i % l == 0) std::cout << "\n";

    std::cout << randomChars[i];

}

std::cout << "\n\nDo you want to save the strings to a text file?\n"
    << "[01] Yes\n"
    << "[02] No\n";

int * choiceFile = new int();
std::cin >> *choiceFile;

if (*choiceFile == 1)
{
    writeToFile(n, l, randomChars);
    std::cout << "\nSaved to 'output.txt'\n" << "Exiting\n";
}
else std::cout << "\nExiting\n";

delete choiceFile;

}
else std::cout << "Invalid input";

}

```