CSI 2372 – Lab Task 4

Abdorrahim Bahrami

Operator Overloading in C++

Your task in this lab is to continue working on operator overloading in C++. This time, you are going to work on more arithmetic operators, and const operators. Make sure you have C++ installed, and you are familiar with the header files, and coding files. If you need help, ask your TA to help you with this.

Then, you should do the following programming task. Each programming task in the lab is a design based on the subjects you learned during lectures. There is a test code that you can use to test your design. If you have questions, ask your TAs.

Most random generators start from a seed or some seeds to generate a sequence of random numbers. Design a class that represents a sequence of a random numbers. The method that we use for generating random numbers use two seeds and it works as follows. The first two numbers in the random numbers sequence are given, which are the two seeds. To generate the next random number, we calculate the multiplication of the two previous numbers and keep the s digits in the middle. If the multiplication does not have $2s$ digits, you should add leading 0s. Your class has s as one of the private members.

$X_0 = 7821$

$X_1 = 2453$

$X_2 = 1849$ $\quad 7821 \times 2453 = 19\mathbf{1849}13$

$X_3 = 5355$ $\quad 2453 \times 1849 = 04\mathbf{5355}97$

$X_4 = \cdots$

Your class must have the following methods. Use the name as they are in the table to be able to use the test file for testing your class design.

You are not allowed to use vector in STL. Use dynamic memory. Fxxking Stupid!

Class RandomGenerator	
Method	**Description**
RandomGenerator	The default constructor that initializes s to be 4 and generate two random 4-digit integers. You can use rand() function in C++ to generate the first two random numbers.
RandomGenerator	The user constructor that receives three integers, which is half of s, the first number is the sequence, and the second number in the sequence. If any of numbers are not s-digit numbers, again you can use rand() function in C++ to generate them.
RandomGenerator	The copy constructor
~ RandomGenerator	The destructor
operator [ ]	For indexing the generated random numbers
operator ==	Checks if two sequence of random numbers are the same
operator !=	Checks if the two sequence of random numbers are not the same
operator =	For assigning a sequence to another sequence
operator +	Adds two sequences of random numbers, if one of them has a smaller number of elements, first more random number should be generated to make them equal length
operator -	Subtracts the second sequence of random numbers from the first one, if one of them has a smaller number of elements, first more random number should be generated to make them equal length
operator *	Multiplies two sequences of random numbers, if one of them has a smaller number of elements, first more random number should be generated to make them equal length
operator +=	Add and assign the result
operator -=	Subtract and assign the result
operator *=	Multiply and assign the result
operator ++ (both forms)	Generates the next random number and adds it to the end of the sequence
operator -- (both forms)	Removes the last generated random number from the end of the sequence. Note that the first two numbers cannot be removed from the sequence.
operator <<	For printing all generated random numbers in the sequence in a comma separated format Example: 7821, 2453, 1849, 5355

You can add any method you need.

Note:

$[X_0, X_1, \dots, X_n] + [Y_0, Y_1, \dots, Y_n] = [(X_0 + Y_0)\ mod\ 10^{s+1}, (X_1 + Y_1)\ mod\ 10^{s+1}, \dots, (X_n + Y_n)\ mod\ 10^{s+1}]$

$[X_0, X_1, \dots, X_n] - [Y_0, Y_1, \dots, Y_2] = [abs(X_0 - Y_0), abs(X_1 - Y_1), \dots, abs(X_n - Y_n)]$

As for multiplication, we do as we did for generating random numbers. Namely,

$X_i \times Y_i = keeping\ the\ s\ digits\ of\ the\ answer\ in\ the\ middle$

Example:

$X_i = 8912$

$Y_i = 1002$

$X_i \times Y_i = 2083 \qquad 8192 \times 1002 = 08208384$

Generating a random number between A and B in C++.

You have to include <cstdlib> for rand().

r = (rand() % (B - A)) + A;