CSE 330 LABORATORY -- Week 1, Spring 2018

Instructor: Kerstin Voigt

This is a "warm-up" and exploratory first CSE 330 lab. Work on the exercises in the order given, and see how far you get. If you manage to complete exercises 1-3, you are in good shape. Exercise 4 (the most interesting one) is left in case there is time left. There is a possibility that it will show up in an upcoming homework assignment. So if you manage to solve, you will be ahead …

Exercise 1: Write a C++ program that will take a positive integer N as input and then print out a matrix of the following sort:

For input 8:

```
1 _ _ _ _ _ _ _
_ 2 _ _ _ _ _ _
_ _ 3 _ _ _ _ _
_ _ _ 4 _ _ _ _
_ _ _ _ 5 _ _ _
_ _ _ _ _ 6 _ _
_ _ _ _ _ _ 7 _
_ _ _ _ _ _ _ 8
```

Exercise 2: If your solution to Exercise 1 is code that is wrapped within "one big main()", unravel your program by having a function print out the matrix. Modify your main() function so that it uses the new function.

(If you already opted for a function implementation in Exercise 1, you are done with Exercise 2 already.)

Exercise 3: Define and implement a class SquareNumber which represents only those numbers that are squares of integer values. Start your class with

```
class SquareNumber
{
public:
      SquareNumber()
            :base(0)
      {}

      // … fill in …

private:
      int base;
};
```

Your class should be written so that it supports the following int main() function:

```
int main()
{
	SquareNumber mysq;
	for (int i = 1; i <= 5; i++)
	{
		mysq.nextsq();
		cout << mysq.getsq() << " ";
	}
	cout << endl;

	SquareNumber sq2(mysq);
	for (int i = 1; i <= 5; i++)
	{
		cout << sq2.getsq() << " ";
		sq2.prevsq();
	}
	cout << endl;

	mysq = sq2;
	for (int i = 1; i <= 10; i++)
		mysq.nextsq();
	cout << mysq.getsq() << endl << endl;


	return 0;
}
```

Test with this int main().

Exercise 4: Write a C++ program that determines the "maximal subsequence sum" of a vector of positive and negative integers. A "subsequence sum" is the sum of any number of contiguous integers in a vector. For example, a vector containing integers

5 -3 7 4 12 -6

… has among its numerous subsequences the sums

2 -- from: 5 – 3 = 1
20 -- from: -3 +7+4+12=20
10 -- from: 4 +12-6 = 10
Etc.

Among all possible subsequence sums, there is one that is maximal. Your program is to compute this value. Start by sketching out an algorithm (go with a simple, straightforward one), before you write any code.

Test with a simple main that reads 10 integers, and computes and prints out the maximal subsequence sum. When you test your program, make sure that you enter a mix of positive and negative numbers (why?).

The instructor and/or lab assistant will make a pass through all lab stations and mark down the progress made by each student. Expect this to happen around 2:30pm, when the lab should be in full swing… No grades will be assigned, but everyone should have something to show or ask important questions.

To obtain credit for this lab: You need not submit anything for this first lab. However, make sure that you will have signed up at **a signup sheet** that will circulate towards the end of the lab session.