LAB EXERCISE

FunLoops

Background:

1. Magic square problem:

   a. Some perfect squares have unique mathematical properties. For example, 36 is:

   - a perfect square, 6^2
   - and the sum of the integers 1 to 8 (1+2+3+4+5+6+7+8 = 36)
   - so let us call a "magic square" any number that is both a perfect square AND equal to the sum of consecutive integers beginning with 1.

   b. The next magic square is 1225:

   - $35^2 = 1225$
   - 1225 = sum of 1 to 49

   c. Write a method that prints the first n magic squares.

2. Reversing an integer problem:

   a. Write a method that reverses the sequence of digits in an integer value.

   - 123 ---> 321
   - 1005 ---> 5001
   - 2500 ---> 52 {you will not have to print out any leading zeroes, such as 0052}

3. Least Common Multiple problem:

   a. write a method that determines the Least Common Multiple of two integers. For example, the LCM of the following pairs:

2,3	LCM = 6
4,10	LCM = 20
12,15	LCM = 60
7,70	LCM = 70

Assignment:

1. Code separate methods to solve each problem.
2. Test each method using the following instructions.
3. You will need to work with long integers for problems 1 & 2.

Instructions:

1. Find the first four magic squares. The first one is the integer 1.

2. Solve these values for reverse the digits:

   12345 10001 1200 5

3. Find the LCM of the following pairs of values:

   15, 18
   40, 12
   2, 7
   100, 5

4. You may use the following form for the `main` method

```
public static void main(String[] args)
{
  FunLoops fun = new FunLoops();

  fun.magicsquare(4);
  System.out.println("12345 reversed ---> " + fun.reverse (12345));
  System.out.println("10001 reversed ---> " + fun.reverse (10001));
  System.out.println("1200 reversed ---> " + fun.reverse (1200));
  System.out.println("5 reversed ---> " + fun.reverse (5));
  System.out.println();
  System.out.println("LCM (15,18) = " + fun.lcm (15,18));
  System.out.println("LCM (40,12) = " + fun.lcm (40,12));
  System.out.println("LCM (2,7) = " + fun.lcm (2,7));
  System.out.println("LCM (100,5) = " + fun.lcm (100,5));
}
```

5. Try finding the first 10 magic squares. This is one of the few programs in this course that take so long that you can accurately time them. Experiment with this program by running it on computers with different clock speed (number of megahertz).