Assignment 1: Winning Lotto!

For this assignment, you will be building your first java application from scratch. You need to remember what you have learned in class, lab, books and your assignments. Be sure to refer to them when you need to.

There are 2 parts to this assignment. In the first part, you are going to be given a problem and you will then need to write an algorithm to solve it. In the second part, you’ll be turning this algorithm into a java program.

So let’s get started!

**Part 1: Picking those winning lottery numbers!**

Your mom loves playing the lottery each week, but she always takes *forever* to pick a set of numbers. It’s driving you crazy because, who does she send to the store to get her lottery tickets? You! Wonderful child that you are, you’ve decided to help her by writing a program that will pick some numbers for her. Awesome!

Think about how you can do this.

* Your mom likes to play both Fantasy 5 and the regular Lotto (6 numbers), so you want to write a program that takes this in account.
  + For Fantasy 5, the range of possible numbers should be from 1 – 36
  + For the Lotto, the range of possible numbers should be from 1 - 53
* Don’t worry about whether you get duplicate numbers (we haven’t learned enough about that yet)
* Assume that you will be given a method that gives you one random number.
* Think about how you want to present her number picks to her. You’ll want to be sure to include some enthusiastic text along with her numbers (e.g., “Here are your winning numbers!”)

*For Part 1*, write an algorithm for your winning lottery number program, and then do several iterations of tests (i.e., step through your algorithm to make sure that it is logically correct and gives you the correct output). Put this in a Word or Open Office document. You’ll turn that document in with the program that you create in Part 2.

**Part 2: Creating your winning lottery numbers program**

Once you are done writing and testing your algorithm, you are ready to start coding!

1. First you need to create a project. Here’s a nice tutorial on how to do that in Netbeans. If you are using Dr. Java or Eclipse, just do a quick search on youtube.com and you’ll find lots of candidates.

<http://www.youtube.com/watch?v=ezUHG1cuxkM>

Be sure to give your project a *nice, meaningful name* (and make sure it adheres to Java’s naming conventions).

1. Once you have your shell ready, there are a few things to know before you start translating your algorithm into code
   * At the top of your class file, be sure to include the following:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// PantherID: [Your PantherID]

// CLASS: COP 2210 – [Semester Year]

// ASSIGNMENT # [#]

// DATE: [Date]

//

// I hereby swear and affirm that this work is solely my own, and not the work

// or the derivative of the work of someone else.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* + Remember I told you that you would be given a method that generates a random number? Here’s what you need to do to use it:
    1. Include the following code at the top of your class file (so that you can use this class:

import java.util.Random;

To find out more about this, go to <http://java.sun.com/javase/7/docs/api/index.html> (like you did in Lab Assignment 2)

* + 1. You’ll need to use some variables. Here’s how you get a random number:

Upper limit of the random number generated (exclusive)

Random r = new Random();

int x = 1 + r.nextInt(10);

Note that the number in the parens (e.g., 10 above) is the upper limit of the random number, exclusively. So, the random number that you get here will be an integer between 1 and 10, once you add the 1. Need a larger range? Just change the 10 to the top of your range.

Here’s another example, in this case if you are printing a random number to the console:

System.out.print( 1 + r.nextInt(5) + " " );

1. Now start translating your algorithm into java code.
   * Remember to code and then compile frequently. It will make it easier to find any bugs.
   * At this point, you can just put everything in the main method. You can use more than one method if you want and it makes sense with your approach, but you are not required to.
2. Once you get your program running correctly, there is one more thing to do. *Any input requested from the user and/or output received from the user should be in a window* (see E.1.14 and E.1.15 from last week’s lab). At this point, you probably have your output going to the console. For your final submission, it needs to go to a window (JOptionPane). Don’t forget any additional libraries that you need to import to do this.

That’s it! *Now show it to your mom (or someone else who likes playing the lottery) and see how impressed she are with your ingenuity*. Of course, you’ll also need to turn it in to Moodle.

**Submission Requirements**

You must upload a zip file to Moodle that includes your complete source project in Netbeans, ready to load, and also contains the output in separate data files, and your Word/Open Office document with your algorithm.

**VERY IMPORTANT:** If you do not provide output in separate, easy to find data files, I will assume that your program does not work on those test cases, and grade accordingly. *Do not embed the output in your source code.*