Java Programming (JP2)

Laboratory Sheet 5

This Lab Sheet contains material based on Lectures 1 – 10 (*up to 23 October 2013*), and contains the submission information for Laboratory 5 (week 6, *28 October – 1 November 2013*).

Be sure to look over the material of Lectures 9 - 10 before Laboratory 5, and bring this sheet to that Laboratory.

You are expected to begin work on laboratory sheets before your scheduled session. This will be necessary if you are to make best use of the presence of your tutor during the lab, and to make a satisfactory attempt at the submission exercise(s).

The deadline for submission of the lab exercise is 24 hours after the end of your scheduled laboratory session in week 6 *(28 October –1 November 2013)*.

Of course you may submit work that is incorrect or incomplete. In order to stretch the stronger members of the class, some of the laboratory exercises are quite challenging, and you should not be too discouraged if you cannot complete all of them.

Aims and objectives

- Class concepts in Java
- Using arrays in Java
- Consolidation of earlier Java concepts

Set up

When you download Laboratory5.zip from moodle, please unzip this file. You will obtain a folder Laboratory5, containing a subfolder entitled Submission5_1. Remember that for this Laboratory you will have to switch your Eclipse workspace to the Laboratory5 folder.

In the folder Submission5_1 will be a file LotterySimulation.java that contains a complete `main()` method, together with a second file, LotteryNumbers.java, that contains a skeleton class. In Eclipse, you should create a new project entitled Submission5_1; the given files will automatically become part of this project.

Submission material

Preparatory work for this programming exercise, prior to your scheduled lab session, is expected and essential to enable you to submit a satisfactory attempt.

Submission exercise 5

The Glasgow University Lottery draw consists of 6 distinct numbers in the range 1 to 49 (inclusive). Participants buy tickets, each ticket holding likewise a set of 6 distinct numbers in the same range. When the draw is made, there are four categories of winning tickets, as follows:

- jackpot winners – all six numbers match the drawn numbers
- runners up – exactly five of the numbers match
- third prize winners – exactly four of the numbers match

- consolation prize winners – exactly three of the numbers match

All other tickets are losers. So, for example, if the draw were to be (5, 15, 30, 35, 40, 48) then ticket (5, 15, 17, 30, 35, 48) would be a runner up, ticket (9, 10, 15, 35, 48, 49) would be a consolation prize winner, and ticket (15, 20, 25, 30, 35, 45) would be a loser.

A LotterySimulation Java program has been written that is designed to

- simulate a lottery draw;
- output the draw – the numbers in the draw should appear in sorted order;
- read in a sequence of tickets from standard input – each ticket is a line of text containing six numbers, in *ascending* order, separated by spaces;
- for each ticket, report its prize status – i.e., whether it is a winner, and if so in what category.

You are provided with the complete code for the class containing the `main()` method – see the code listing overleaf. In fact this program does a little more than is described above – for test purposes, it also generates a 'fixed' draw and reports the prize status of each ticket relative to this draw also.

For the scenario described above, the input to the program would be:

```
 5  15  17  30  35  48
 9  10  15  35  48  49
15  20  25  30  35  45
```

and the output would be:

```
The random lottery draw:     5  15  30  35  40  48
The fixed lottery draw:      5  10  15  20  25  30


Ticket:  5  15  17  30  35  48   runner up
Ticket:  5  15  17  30  35  48   consolation

Ticket:  9  10  15  35  48  49   consolation
Ticket:  9  10  15  35  48  49   loser

Ticket: 15  20  25  30  35  45   third prize
Ticket: 15  20  25  30  35  45   third prize
```

The first line of output for each ticket is relative to the random draw and the second line is relative to the fixed draw.

The program uses a class called `LotteryNumbers`, which can be thought of as representing a Lottery ticket or a Lottery draw – essentially a sequence of six distinct integers in the appropriate range. You are to implement this class in such a way that the main program has the required functionality. ***You should not change in any way the `LotterySimulation` class.*** You may verify the functionality of your `LotteryNumbers` implementation by running the `LotterySimulation` or the `TestLotteryNumbers` JUnit test file.

Hints

1. The input may contain an *arbitrary* number of lines. The program should **not** prompt the user to provide input – when the program is tested in JUnit, standard input will be re-directed to come from preset test data.

2. You can deduce from the main program the names and functionality of some public methods that must be defined in the LotteryNumbers class. You should try to identify any additional helper methods that may be appropriate – for example, a method to determine the number of 'hits' between a given set of lottery numbers and another set.

3. You need provide in the LotteryNumbers class only methods that are required in the context of this program. There is no need to include methods that are not used, even standard methods (accessors, mutators, etc.) that you would normally expect to provide in a class.

3. You will have to figure out a way to use the Math.random() method to generate six numbers in the required range. You will need some mechanism to ensure that no **repeated** numbers appear in the draw.

Submission

You should submit your work before the deadline no matter whether the programs are fully working or not.

When you are ready to submit, go to the JP2 moodle site. Click on Laboratory 5 Submission. Click 'Add Submission'. Open Windows Explorer and browse to the folder that contains your Java source code ...\Laboratory5\Submission5_1\ and drag *only* the *single* Java file LotteryNumbers.java into the drag-n-drop area on the moodle submission page. **Your markers only want to read your java file, not your class file.** Then click the blue save changes button. Check the single .java file is uploaded to the system. Then click submit assignment and fill in the non-plagiarism declaration. Your tutor will inspect your file and return feedback to you via moodle.

Gaining the credits for JP2

Recall that the credit criteria for JP2 include obtaining at least 7 ticks for lab assignments. To obtain a tick you must attend the lab and submit the assignment.

The class containing the main program

```
import java.util.Scanner;

public class LotterySimulation {

   public static void main(String[] args) {

      Scanner inputScanner = new Scanner(System.in);

      // make and report the draw
      LotteryNumbers randomDraw = LotteryNumbers.makeDraw();
      System.out.println("\nThe random lottery draw: " + randomDraw + "\n");

      // fix a draw for test purposes
      LotteryNumbers fixedDraw = new LotteryNumbers(" 5 10 15 20 25 30");
      System.out.println("\nThe fixed lottery draw: " + fixedDraw + "\n");

      // now process the tickets
      while (inputScanner.hasNextLine()){

         String nextLine = inputScanner.nextLine();
```

```
        // quit if line is empty
        if (line.length() == 0) break;

        LotteryNumbers ticket = new LotteryNumbers(nextLine);

        System.out.println(ticket.status(randomDraw));
        System.out.println(ticket.status(fixedDraw));
        System.out.println();
     }

   }

}
```