CS 2, Fall 2015 **CS 2 Cover Letter**
Dr. Gurka

Name: Sergiy Kolodyazhnyy
Assignment: Project #3 Lotto
Date submitted: 9 / 16 / 2015
Total time: about 30 - 34 hours

On time or late? On time
GOOJF? no

Did you collaborate with any classmates on this project? No
If yes, who and what did you work together on?

Did you get any tutoring or similar help on this project? Explain.

Mostly referenced to java API documentation online.

How'd it go?

What went well?
The output of the program in the file shows total number of runs close to 5 million, which is what is close to instructor's data, as well as correlated with probability of winning in such lottery
What problems did you have?
Mostly with properly passing variables between methods as well as keeping track of numerous variables
What did you learn new?

It's somewhat hard to pinpoint the newly learned things, since for the most part we've been working with the structures we've used before – for, while, nested for-while loops, PrintWriter and FileWriter classes. It wasnt required in the project but I did practice creating an accessor method as well as worked on creating a simple constructor.

Any remaining questions?

Other comments on the project?

The small observation I've noticed is that outputing text to console greatly increases execution time. The less output to the console, the faster the program performed. On average, Unix's /usr/bin/time reported approximately 40 seconds to run 10 games. Of course it doesn't subtract taking input from the user, so the actual performance of the Lotto class may well be even better.

If this project is late, what is not included or not working correctly?

Other discussions as specified with the project.

Driver Class

```
 1: /**************************************
 2: Author: Sergiy Kolodyazhnyy
 3: Course: CS 2050
 4: Date: Sept 16 2015
 5: Instructor: Prof Gurka
 6: Java version: OpenJDK, 1.7.0
 7: IDE: nano text-editor and java compiler
 8: **************************************/
 9: import javax.swing.JOptionPane;
10: import java.io.PrintWriter;
11: import java.io.FileWriter;
12: import java.io.IOException;
13:
14: public class cs2hw3Driver
15: {
16:     public static void main (String[] args) throws IOException
17:     {
18:
19:         String outputFilePath = getFileName();
20:     FileWriter outputFile = new FileWriter(outputFilePath);
21:     PrintWriter writeToFile = new PrintWriter(outputFile);
22:         int numsArr [] = getNums( );
23:     cs2hw3Lotto Lotto  = new cs2hw3Lotto (numsArr);
24:
25:     int [] statistics = new int[5];
26:     int[] sums = new int[5];
27:     printHeader(writeToFile);
28:     for (int x = 1; x<=10; x++)
29:     {     statistics = Lotto.playTillJackpot();
30:           printToFile(writeToFile,statistics,x);
31:           for (int y = 0; y<5; y++)
32:           {
33:                 sums[y] += statistics[y];
34:           }
35:     }
36:     printAverages(writeToFile,sums);
37:     writeToFile.close();
38:
39:     }
40: //---------------------------------------------------------------------
41:     public static int[]  getNums(  )
42:     {
43:     // had to declare instance of the Lotto object here to use
44:     // hasDuplicates method
45:     cs2hw3Lotto Lottery = new cs2hw3Lotto();
46:
47:         String numString = "";
48:     int arr[];
49:     // loop until the user gives right input
50:     while ( true )
51: {   numString = JOptionPane.showInputDialog("Please input 6 unique digits , from 1
to 41, comma-separated");
52:     // check if we have illegal characters with regex patter matching
53:         if(numString.matches(".*[^0-9,].*"))
54:          {
55:                 System.out.println("REgex works");
56:                 continue;
57:          }
58:
59:     // protective feature - removing stray blanks from user's input
60:         numString = numString.replaceAll("\\s+","");
```

```
    61:
    62:
    63:     // split the string into array of strings using comma as delimiter
    64:         String [] numArrStr = numString.split(",");
    65:     //declare int array of same size as numeric string array
    66:         arr = new int[numArrStr.length];
    67:     // check if the length is wrong (meaning user missed comma or put too many
nums)
    68:     if ( numArrStr.length < 6 || numArrStr.length > 6)
    69:           continue;
    70:     // If above conditions are OK, fill the array of ints by parsing array
    71:     // of strings to int
    72:         for (int i = 0 ; i < numArrStr.length; i++)
    73:             {
    74:                 arr[i] = Integer.parseInt(numArrStr[i]);
    75:             }
    76:
    77:     // Check if we have duplicate numbers
    78:     if ( Lottery.hasDuplicates(arr))
    79:     {     continue;
    80:     }
    81:     else
    82:     {     break;
    83:     }
    84:
    85: }
    86:         return arr;
    87:     }
    88: //=================================================================
    89:     // as the name suggests, here we are getting the filename
    90:     // where the output will be stored
    91:     public static String getFileName ()
    92:     {
    93:         return JOptionPane.showInputDialog("Please input filename where data will
be stored");
    94:     }
    95:
    96: //=================================================================
    97: // the three methods bellow output to the file.  printToFile is called repeatedly
    98: // while printHeader and printAverages serve purpose at the beginnning and ending
    99: // of the 10 games
   100:     public static void printToFile( PrintWriter statsFile,int[] stats, int
game   )
   101:     {
   102:           statsFile.printf("%-2d: %-10d %-10d %-10d %-10d %-
10d\n",game,stats[0],stats[1],stats[2],stats[3],stats[4]);
   103:     }
   104:
   105:     public static void printHeader( PrintWriter statsFile  )
   106:     {
   107:           statsFile.printf("%s %-10s %-10s %-10s %-10s %-
10s\n","Game#","Rolls","Match 3","Match 4","Match 5","Payout");
   108:
   109:     }
   110:     public static void printAverages( PrintWriter statsFile, int sums [] )
   111:     {
   112:       statsFile.println("Average values:");
   113:      for (int i=0; i<5;i++)
   114:       statsFile.printf("\t%-10d",sums[i]/10);
   115:     }
   116: }
```

Lotto Class

```
 1: import java.lang.Math;
 2: import java.util.Random;
 3: import java.util.Arrays;
 4:
 5: public class cs2hw3Lotto
 6: {
 7:     // MAX and MIN are made private and final, because
 8:     // we want these to remain constant and unalterable
 9:     private final int MAX =  41;
10:     private final int MIN = 1;
11:
12:     int [] userNums;
13: //  int stats [] = new int[5];
14:
15:     Random rand = new Random();
16:
17:     public cs2hw3Lotto()
18:     {
19:
20:     }
21:
22:     // Constructor for our Lotto object
23:     public cs2hw3Lotto (int [] a)
24:     {
25:       userNums = a;
26:     }
27:
28:     public void checkUserInput()
29:     {
30:           for (int i = 0 ; i < userNums.length; i++)
31:                 System.out.print(userNums[i]);
32:     }
33: //=========================================================================
34:     // random number generator method
35:     public int getRandomInt ()
36:     {
37:           int out;
38:           out = rand.nextInt((MAX - MIN) + 1) + MIN;
39:           return out;
40:           }
41:
42: //=========================================================================
43:     // a helper method that tests whether an array has duplicates
44:     // will be used to test user's input as well as
45:     // the generated numbers
46:     public boolean hasDuplicates (int[] a)
47:     {
48:           boolean result = false;
49:           Arrays.sort(a);
50:
51:           for(int i = 1; i < a.length; i++)
52:             {
53:                 if(a[i] == a[i - 1])
54:                     {
55:                         result = true;
56:                     }
57:             }
58:           return result;
59:
60:           }
61: //=========================================================================
```

```
    62:
    63:     public int [] playTillJackpot()
    64:     {
    65:           // stats array hold total number of plays,
    66:           // how many times we matched 3 numbers,
    67:           // 4 numbers, 5 numbers, and total payouts in
    68:           // that order respectivelly
    69:           int stats [] = new int[5];
    70:           int countMatched = 0;
    71:           int lottoNums [] = new int[6];
    72:           while (true)
    73:                     {
    74:                 // generate numbers
    75:                         for (int i = 0; i < 6; i++)
    76:                             {
    77:                                 lottoNums[i] = getRandomInt();
    78:                             }
    79:                 // re-run number generator if we have duplicate numbers
    80:                         if (hasDuplicates(lottoNums))
    81:                             {
    82:                                 continue;
    83:                             }
    84:                 stats[0] ++ ;
    85:                         //compare userNums to lottoNums
    86:                         // count matched numbers
    87:                         for (int j = 0; j < 6; j++)
    88:                             {
    89:                                 for (int k = 0 ; k < 6; k++)
    90:                                     {
    91:                                         if (lottoNums[k] == userNums[j])
    92:                                             countMatched++;
    93:                                     }
    94:                             }
    95:                 switch (countMatched)
    96:                 {
    97:                       case 3: stats[1]++;break;
    98:                       case 4: stats[2]++;break;
    99:                       case 5: stats[3]++;break;
   100:                       default: break;
   101:                 }
   102:
   103:                         if (countMatched == 6)
   104:                             {
   105:                                // System.out.println("Jackpot");
   106:                       //System.out.println("Played " + stats[0] + " times");
   107:                       stats[4] = stats[1]*10 + stats[2]*50 + stats[3]*1000;
   108:                       //System.out.println("Payout " + stats[4]);
   109:                                 break;
   110:                             }
   111:                         countMatched = 0;
   112:
   113:     }
   114:   return stats;
   115:   }
   116: }
```

Output file

```
Game# Rolls      Match 3    Match 4    Match 5    Payout
1 : 7366026    213667     14760      313        3187670
2 : 7027934    204471     14053      285        3032360
3 : 5360917    155998     10742      228        2325080
```

```
4 : 10643778    309764      21003       504         4651790
5 : 3908225     114075      7661        152         1675800
6 : 2347405     68555       4722        114         1035650
7 : 2182544     63529       4296        110         960090
8 : 680140      19766       1333        36          300310
9 : 5395723     156513      10877       251         2359980
10: 1865254     54530       3782        84          818400
Average values:
      4677794     136086      9322        207         2034713
```

Program behavior when prompted for input