

1a	<pre>private static int numberOfLeapYears(int year1, int year2){ int counter = 0; for(int i = year1; i<=year2; i++){ if(isLeapYear(i)) counter++;} return counter; }</pre>
1b	<pre>public static int dayOfWeek(int month, int day, int year) { return (firstDayOfYear(year)+dayofYear(month, day, year)-1)%7; }</pre>

Rubric 1a	
+1	Creates an int counter variable to track results of leap years
+1	Creates a for loop without out of bounds exception that includes year1 and year 2
+1	Correctly calls isLeapYear function
+1	Correctly increments counter if a year is a Leap Year
+1	Returns correct count of number of Leap Years between year1 and year2
Rubric 1b	
+1	Correctly calls firstDayOfYear function
+1	Correctly calls dayofYear function
+1	Calculates the correct day of the week using %7
+1	Returns correct dayOfWeek int

```

2 public class StepTracker{
    private int minSteps;
    private ArrayList<Double> activeSteps = new ArrayList<Double>();

    public StepTracer(int minSteps){
        this.minSteps = minSteps;}

    public void addDailySteps(int steps){
        activeSteps.add(steps);}

    public int activeDays(){
        int counter = 0;
        for(Double a: activeSteps){
            if(a>=minSteps) counter++;}
        return counter;}

    public double averageSteps(){
        double average = 0.0;
        if(activeSteps.size()==0) return average;
        for(Double a: activeSteps){
            average+=a;}
        return average/activeSteps.size();}
}
**There were many ways to do this without creating an ArrayList!

```

Rubric 2	
+1	Correctly creates instance variables that are set to private (could do 4 variables to track – int minimumSteps, int ActiveDays, int TotalDays, double stepCount) or what I've shown in my solution.
+1	Creates a constructor that has the correct parameter of an int that it correctly sets the minimumSteps instance variable.
+1	public void addDailySteps(int steps) is the method name.
+1	addDailySteps correctly increments the TotalDays, stepCount correctly
+1	public int activeDays() declared correctly
+1	public int activeDays() returns the correct count of active days *it's likely increasing your active days is in addDailySteps, which is needed for this point!
+1	public double averageSteps() declared correctly
+1	averageSteps does not attempt to divide by 0 and returns a 0.0 in the event that no steps were recorded
+1	public double averageSteps() returns the correct double value that represents the average steps (all steps added together divided by total days)

3a	<pre> public ArrayList<String> getDelimitersList(String[] tokens) { ArrayList<String> delimiters = new ArrayList<>(); for(String a: tokens) if(a.equals(openDel) a.equals(closeDel)) delimiters.add(a); return delimiters; } </pre>
3b	<pre> public static boolean isBalanced(ArrayList<String> delimiters) { int open = 0; int close = 0; for(String a: delimiters){ if(a.equals(openDel)) open++; else if(a.equals(closeDel)) close++; if(close>open) return false; } return open==close; } </pre>

Rubric 3a	
+1	Instantiate an arraylist of Strings correctly
+1	Create a for loop that goes through every entry in the array tokens without out of bounds exception
+1	Correctly tests if the item in the tokens array is an open or closed delimiter
+1	Returns an arraylist with ONLY the correct delimiters in it
Rubric 3b	
+1	Creates a for loop that correctly goes through all items in the delimiters arraylist with no out of bounds exceptions
+1	Correctly increments conditions that add 1 to variables counting the closed and open delimiters
+1	Correctly checks if closed<open inside for loop and returns false if so
+1	Correctly checks if closed and open values are equal after loop is executed
+1	Returns correct true/false value without error

4a	<pre>public LightBoard(int numRows, int numCols){ lights = new boolean[numCols][numRows]; for(int i = 0; i<numRows; i++){ for(int j=0; i<numCols; j++){ if(Math.random(<0.4) lights[i][j] = true; } } }</pre> <p>*note I didn't put a condition for lights[i][j]=false because the array will default to those values, but it is not wrong if you did</p>
4b	<pre>public boolean evaluateLight(int row, int col) { boolean a = lights[row][col]; int count = 0; for(int i = 0; i<lights.length;i++){ if(lights[i][col]) count++;} if(a && count%2==0) return false; else if(!a && count%3==0) return true; return a; }</pre>

Rubric 4a	
+1	Instantiates the lights array with correct row and columns
+1	Creates loops that correctly go through all items in the 2D array without out of bounds exception
+1	Creates a random number generator correctly that can be used to represent the simulation correctly in the loop (so it will generate each increment of the loop)
+1	Correctly assigns true or false to each item in the 2D arrayList
Rubric 4b	
+1	Creates for loop to go through contents of the column specified in the parameters
+1	Correctly adds the number of lights on in the column
+1	Returns false if the light in the row&&column is on and the column count of lights on is even
+1	Returns true if the light in the row&&column is off and the column count is divisible by 3
+1	Returns the correct true/false condition in all cases (if none of the above conditions are met, returns if light at row&&column is on or off)