

## 9 points

## Question 4: 2D Arrays

## Canonical solution

a. `public SumOrSameGame(int numRows, int numCols)` **4 points**

```

{
    puzzle = new int[numRows][numCols];

    for (int row = 0; row < numRows; row++)
    {
        for (int col = 0; col < numCols; col++)
        {
            puzzle[row][col] = (int)(Math.random() * 9) + 1;
        }
    }
}

```

b. `public boolean clearPair(int row, int col)` **5 points**

```

{
    int val1 = puzzle[row][col];
    for (int currRow = row; currRow < puzzle.length; currRow++)
    {
        for (int currCol = 0; currCol < puzzle[0].length; currCol++)
        {
            int val2 = puzzle[currRow][currCol];
            if (currRow != row || currCol != col)
            {
                if (val1 == val2 || val1 + val2 == 10)
                {
                    puzzle[row][col] = 0;
                    puzzle[currRow][currCol] = 0;
                    return true;
                }
            }
        }
    }
    return false;
}

```

## a. SumOrSameGame

| Scoring Criteria |                                                                                                              | Decision Rules                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |
|------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 1                | Constructs correctly sized 2D array of <code>int</code> and assigns to instance variable <code>puzzle</code> | Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>fail to update the instance variable <code>puzzle</code>, e.g., by redeclaring as a local variable</li> <li>print or return a value instead of or in addition to updating <code>puzzle</code></li> </ul>                                                                                                                                                                                                                                                                                                                                            | 1 point |
| 2                | Traverses 2D array ( <i>no bounds errors</i> )                                                               | Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>fail to construct the 2D array, as long as traversal uses correct bounds</li> <li>fail to assign to the 2D array elements</li> </ul> <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>fail to access an element of the 2D array in the traversal</li> </ul>                                                                                                                                                                                                                                     | 1 point |
| 3                | Generates a random integer that is uniform in the range <code>[1, 9]</code>                                  | Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>fail to call <code>Math.random</code> or an equivalent method</li> <li>make any incorrect call to <code>Math.random</code> or an equivalent method</li> <li>call <code>random</code> without <code>Math.</code></li> <li>fail to cast to an <code>int</code></li> </ul>                                                                                                                                                                                                                                                                             | 1 point |
| 4                | Assigns values to 2D array elements ( <i>algorithm</i> )                                                     | Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>generate the random value incorrectly</li> <li>assign to a local 2D array instead of the instance variable</li> <li>fail to assign some elements due to a bounds error</li> </ul> <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>assign a value other than the attempted randomly generated value</li> <li>correctly initialized the 2D array but now assign somewhere else</li> <li>assign the same value to all visited elements</li> <li>fail to assign to all visited elements</li> </ul> | 1 point |

b. `clearPair`

| Scoring Criteria            |                                                                                                                                   | Decision Rules                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |         |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 5                           | Accesses all and only necessary elements of <code>puzzle</code> in rows <code>&gt;= row</code> ( <i>no bounds errors</i> )        | <p>Responses <b>can</b> still earn the point even if they</p> <ul style="list-style-type: none"> <li>access additional rows, as long as they also guard against pairing them</li> <li>pair the element at <code>row</code> and <code>col</code> with itself</li> <li>omit extra elements due to incorrect self-pairing guard, as long as the bounds would otherwise support accessing all necessary elements</li> <li>return early, as long as bounds and indices would otherwise support accessing all necessary elements</li> </ul> <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>fail to access elements of <code>puzzle</code> correctly</li> </ul> | 1 point |
| 6                           | Guards against pairing an element with itself                                                                                     | <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>omit extra potential pairs with their self-pairing guard</li> <li>have no loop</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1 point |
| 7                           | Determines whether two elements are the same or sum to 10                                                                         | <p>Responses <b>can</b> still earn the point even if they</p> <ul style="list-style-type: none"> <li>pair the element at <code>row</code> and <code>col</code> with itself</li> <li>have no loop</li> </ul> <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>test only one of the conditions</li> </ul>                                                                                                                                                                                                                                                                                                                                                    | 1 point |
| 8                           | Sets a 2D array element to 0 ( <i>in the context of a loop</i> )                                                                  | <p>Responses <b>can</b> still earn the point even if they</p> <ul style="list-style-type: none"> <li>set only one identified element to 0</li> <li>incorrectly identify elements</li> <li>set more than 2 elements to 0</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1 point |
| 9                           | Sets identified elements to 0 when match is found and returns appropriate <code>boolean</code> in both cases ( <i>algorithm</i> ) | <p>Responses <b>can</b> still earn the point even if they</p> <ul style="list-style-type: none"> <li>identify elements incorrectly, as long as there is some conditional selection</li> </ul> <p>Responses <b>will not</b> earn the point if they</p> <ul style="list-style-type: none"> <li>set only one identified element to 0</li> <li>change more than 1 element besides the element at <code>row</code> and <code>col</code></li> <li>return an incorrect value due to an early return</li> <li>return an incorrect value due to not returning after clearing once</li> </ul>                                                                                                                      | 1 point |
| Question-specific penalties |                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |         |
| None                        |                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |         |

2019

# AP<sup>®</sup> Computer Science A

## Scoring Guidelines

## 2019 SCORING GUIDELINES

Apply the question assessment rubric first, which always takes precedence. Penalty points can only be deducted in a part of the question that has earned credit via the question rubric. No part of a question (a, b, c) may have a negative point total. A given penalty can be assessed only once for a question, even if it occurs multiple times or in multiple parts of that question. A maximum of 3 penalty points may be assessed per question.

## 1-Point Penalty

- v) Array/collection access confusion (`[] get()`)
- w) Extraneous code that causes side-effect (e.g., printing to output, incorrect precondition check)
- x) Local variables used but none declared
- y) Destruction of persistent data (e.g., changing value referenced by parameter)
- z) Void method or constructor that returns a value

## No Penalty

- o Extraneous code with no side-effect (e.g., valid precondition check, no-op)
- o Spelling/case discrepancies where there is no ambiguity\*
- o Local variable not declared provided other variables are declared in some part
- o `private` or `public` qualifier on a local variable
- o Missing `public` qualifier on class or constructor header
- o Keyword used as an identifier
- o Common mathematical symbols used for operators (`*` `*` `+` `+` `≤` `<` `>` `≠`)
- o `[]` vs. `()` vs. `<>`
- o `=` instead of `==` and vice versa
- o `length/size` confusion for array, `String`, `List`, or `ArrayList`; with or without `()`
- o Extraneous `[]` when referencing entire array
- o `[i,j]` instead of `[i][j]`
- o Extraneous size in array declaration, e.g., `int[size] nums = new int[size];`
- o Missing `;` where structure clearly conveys intent
- o Missing `{ }` where indentation clearly conveys intent
- o Missing `()` on parameter-less method or constructor invocations
- o Missing `()` around `if` or `while` conditions

\*Spelling and case discrepancies for identifiers fall under the “No Penalty” category only if the correction can be **unambiguously** inferred from context, for example, “`ArrayList`” instead of “`Arraylist`”. As a counterexample, note that if the code declares “`int G=99, g=0;`”, then uses “`while (G < 10)`” instead of “`while (g < 10)`”, the context does **not** allow for the reader to assume the use of the lower-case variable.

## 2019 SCORING GUIDELINES

## Question 1: Calendar

|          |                                |          |
|----------|--------------------------------|----------|
| Part (a) | <code>numberOfLeapYears</code> | 5 points |
|----------|--------------------------------|----------|

**Intent:** Return the number of leap years in a range

- +1 Initializes a numeric variable
- +1 Loops through each necessary year in the range
- +1 Calls `isLeapYear` on some valid year in the range
- +1 Updates count based on result of calling `isLeapYear`
- +1 Returns count of leap years

|          |                        |          |
|----------|------------------------|----------|
| Part (b) | <code>dayOfWeek</code> | 4 points |
|----------|------------------------|----------|

**Intent:** Return an integer representing the day of the week for a given date

- +1 Calls `firstDayOfYear`
- +1 Calls `dayOfYear`
- +1 Calculates the value representing the day of the week
- +1 Returns the calculated value

## Question-Specific Penalties

- 1 (t) Static methods called with `this`.

## 2019 SCORING GUIDELINES

## Question 1: Scoring Notes

| Part (a) numberOfLeapYears |                                                       |                                                                                                                               | 5 points                                                                                                                                                        |
|----------------------------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Points                     | Rubric Criteria                                       | Responses earn the point even if they...                                                                                      | Responses will not earn the point if they...                                                                                                                    |
| +1                         | Initializes a numeric variable                        |                                                                                                                               | <ul style="list-style-type: none"> <li>use the variable for loop control only</li> </ul>                                                                        |
| +1                         | Loops through each necessary year in the range        |                                                                                                                               | <ul style="list-style-type: none"> <li>consider years outside the range</li> </ul>                                                                              |
| +1                         | Calls isLeapYear on some valid year in the range      | <ul style="list-style-type: none"> <li>do not use a loop</li> </ul>                                                           |                                                                                                                                                                 |
| +1                         | Updates count based on result of calling isLeapYear   | <ul style="list-style-type: none"> <li>do not use a loop</li> <li>do not initialize the counter</li> </ul>                    | <ul style="list-style-type: none"> <li>use result as a non-boolean</li> </ul>                                                                                   |
| +1                         | Returns count of leap years                           | <ul style="list-style-type: none"> <li>loop from year1 to year2 incorrectly</li> <li>do not initialize the counter</li> </ul> | <ul style="list-style-type: none"> <li>do not use a loop</li> <li>update or initialize the counter incorrectly</li> <li>return early inside the loop</li> </ul> |
| Part (b) dayOfWeek         |                                                       |                                                                                                                               | 4 points                                                                                                                                                        |
| Points                     | Rubric Criteria                                       | Responses earn the point even if they...                                                                                      | Responses will not earn the point if they...                                                                                                                    |
| +1                         | Calls firstDayOfYear                                  |                                                                                                                               | <ul style="list-style-type: none"> <li>do not use the given year</li> </ul>                                                                                     |
| +1                         | Calls dayOfYear                                       |                                                                                                                               | <ul style="list-style-type: none"> <li>have arguments out of order</li> </ul>                                                                                   |
| +1                         | Calculates the value representing the day of the week |                                                                                                                               | <ul style="list-style-type: none"> <li>make any error in the calculation</li> </ul>                                                                             |
| +1                         | Returns the calculated value                          | <ul style="list-style-type: none"> <li>return the value from calling firstDayOfYear or dayOfYear</li> </ul>                   | <ul style="list-style-type: none"> <li>return a constant value</li> </ul>                                                                                       |

## 2019 SCORING GUIDELINES

## Question 1: Calendar

## Part (a)

```
public static int numberOfLeapYears(int year1, int year2)
{
    int count = 0;
    for (int y = year1; y <= year2; y++)
    {
        if (isLeapYear(y))
        {
            count++;
        }
    }
    return count;
}
```

## Part (b)

```
public static int dayOfWeek(int month, int day, int year)
{
    int startDay = firstDayOfYear(year);
    int nthDay = dayOfYear(month, day, year);
    int returnDay = (startDay + nthDay - 1) % 7;
    return returnDay;
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.

## 2019 SCORING GUIDELINES

## Question 2: Step Tracker

|               |             |                 |
|---------------|-------------|-----------------|
| <b>Class:</b> | StepTracker | <b>9 points</b> |
|---------------|-------------|-----------------|

**Intent:** Define implementation of a class to record fitness data

- +1** Declares all appropriate `private` instance variables
- +2** Constructor
  - +1** Declares header: `public StepTracker(int ____)`
  - +1** Uses parameter and appropriate values to initialize instance variables
- +3** `addDailySteps` method
  - +1** Declares header: `public void addDailySteps(int ____)`
  - +1** Identifies active days and increments count
  - +1** Updates other instance variables appropriately
- +1** `activeDays` method
  - +1** Declares and implements `public int activeDays()`
- +2** `averageSteps` method
  - +1** Declares header: `public double averageSteps()`
  - +1** Returns calculated `double` average number of steps

## 2019 SCORING GUIDELINES

## Question 2: Scoring Notes

| Class StepTracker |                                                                        | 9 points                                                                                                                                                                                                |                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Points            | Rubric Criteria                                                        | Responses earn the point even if they...                                                                                                                                                                | Responses will not earn the point if they...                                                                                                                                                                                                                           |
| +1                | Declares all appropriate <code>private</code> instance variables       |                                                                                                                                                                                                         | <ul style="list-style-type: none"> <li>omit keyword <code>private</code></li> <li>declare variables outside the class</li> </ul>                                                                                                                                       |
| +2                | Constructor                                                            |                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                        |
| +1                | Declares header: <code>public StepTracker(int ____)</code>             | <ul style="list-style-type: none"> <li>omit keyword <code>public</code></li> </ul>                                                                                                                      | <ul style="list-style-type: none"> <li>declare method <code>private</code></li> </ul>                                                                                                                                                                                  |
| +1                | Uses parameter and appropriate values to initialize instance variables | <ul style="list-style-type: none"> <li>initialize primitive instance variables to default values when declared</li> </ul>                                                                               | <ul style="list-style-type: none"> <li>fail to use the parameter to initialize some instance variable</li> <li>fail to declare instance variables</li> <li>initialize local variables instead of instance variables</li> <li>assign variables to parameters</li> </ul> |
| +3                | <code>addDailySteps</code> method                                      |                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                        |
| +1                | Declares header: <code>public void addDailySteps(int ____)</code>      | <ul style="list-style-type: none"> <li>omit keyword <code>public</code></li> </ul>                                                                                                                      | <ul style="list-style-type: none"> <li>declare method <code>private</code></li> </ul>                                                                                                                                                                                  |
| +1                | Identifies active days and increments count                            | <ul style="list-style-type: none"> <li>put valid comparison erroneously in some other method</li> </ul>                                                                                                 | <ul style="list-style-type: none"> <li>fail to use the parameter as part of the comparison</li> <li>fail to increment a count of active days</li> <li>fail to increment an instance variable</li> <li>compare parameter to some numeric constant</li> </ul>            |
| +1                | Updates other instance variables appropriately                         |                                                                                                                                                                                                         | <ul style="list-style-type: none"> <li>update another instance variable only on active days</li> <li>update another instance variable inappropriately</li> <li>fail to update appropriate instance variable</li> <li>update a local variable</li> </ul>                |
| +1                | <code>activeDays</code> method                                         |                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                        |
| +1                | Declares and implements <code>public int activeDays()</code>           | <ul style="list-style-type: none"> <li>return appropriate count of active days where the instance variables were updated improperly in <code>addDailySteps</code> or <code>activeDays</code></li> </ul> | <ul style="list-style-type: none"> <li>declare method <code>private</code></li> <li>return value that is not the number of active days</li> <li>fail to return a value</li> </ul>                                                                                      |

## 2019 SCORING GUIDELINES

## Question 2: Scoring Notes (continued)

| Points    | Rubric Criteria                                         | Responses earn the point even if they...                                                                                                                                                      | Responses will not earn the point if they...                                                                                                                 |
|-----------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>+2</b> | <b>averageSteps method</b>                              |                                                                                                                                                                                               |                                                                                                                                                              |
| +1        | Declares header:<br>public double<br>averageSteps()     | <ul style="list-style-type: none"> <li>omit keyword public</li> </ul>                                                                                                                         | <ul style="list-style-type: none"> <li>declare method private</li> </ul>                                                                                     |
| +1        | Returns calculated<br>double average<br>number of steps | <ul style="list-style-type: none"> <li>maintain instance variables improperly but calculate appropriate average</li> <li>fail to handle the special case where no days are tracked</li> </ul> | <ul style="list-style-type: none"> <li>use integer division</li> <li>calculate something other than steps divided by days</li> <li>fail to return</li> </ul> |

## 2019 SCORING GUIDELINES

## Question 2: Step Tracker

```

public class StepTracker
{
    private int minSteps;
    private int totalSteps;
    private int numDays;
    private int numActiveDays;

    public StepTracker(int threshold)
    {
        minSteps = threshold;
        totalSteps = 0;
        numDays = 0;
        numActiveDays = 0;
    }

    public void addDailySteps(int steps)
    {
        totalSteps += steps;
        numDays++;
        if (steps >= minSteps)
        {
            numActiveDays++;
        }
    }

    public int activeDays()
    {
        return numActiveDays;
    }

    public double averageSteps()
    {
        if (numDays == 0)
        {
            return 0.0;
        }
        else
        {
            return (double) totalSteps / numDays;
        }
    }
}

```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.

## 2019 SCORING GUIDELINES

## Question 3: Delimiters

|                 |                                |                 |
|-----------------|--------------------------------|-----------------|
| <b>Part (a)</b> | <code>getDelimitersList</code> | <b>4 points</b> |
|-----------------|--------------------------------|-----------------|

**Intent:** Store delimiters from an array in an `ArrayList`

- +1** Creates `ArrayList<String>`
- +1** Accesses all elements in array `tokens` (no bounds errors)
- +1** Compares strings in `tokens` with both instance variables (must be in the context of a loop)
- +1** Adds delimiters into `ArrayList` in original order

|                 |                         |                 |
|-----------------|-------------------------|-----------------|
| <b>Part (b)</b> | <code>isBalanced</code> | <b>5 points</b> |
|-----------------|-------------------------|-----------------|

**Intent:** Determine whether open and close delimiters in an `ArrayList` are balanced

- +1** Initializes accumulator(s)
- +1** Accesses all elements in `ArrayList delimiters` (no bounds errors)
- +1** Compares strings in `delimiters` with instance variables and updates accumulator(s) accordingly
- +1** Identifies and returns appropriate `boolean` value to implement one rule
- +1** Identifies and returns appropriate `boolean` values for all cases

## 2019 SCORING GUIDELINES

## Question 3: Scoring Notes

| <b>Part (a)</b> <code>getDelimitersList</code> |                                                                                                            | <b>4 points</b>                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                        |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Points                                         | Rubric Criteria                                                                                            | Responses earn the point even if they...                                                                                                                                                                                                                                    | Responses will not earn the point if they...                                                                                                                                                                           |
| <b>+1</b>                                      | Creates <code>ArrayList&lt;String&gt;</code>                                                               | <ul style="list-style-type: none"> <li>omit <code>&lt;String&gt;</code></li> </ul>                                                                                                                                                                                          | <ul style="list-style-type: none"> <li>omit the keyword <code>new</code></li> </ul>                                                                                                                                    |
| <b>+1</b>                                      | Accesses all elements in array <code>tokens</code> (no bounds errors)                                      | <ul style="list-style-type: none"> <li>return incorrectly inside the loop</li> </ul>                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>treat <code>tokens</code> as a single string</li> <li>access elements of <code>tokens</code> as if from an <code>ArrayList</code> (e.g., <code>tokens.get(i)</code>)</li> </ul> |
| <b>+1</b>                                      | Compares strings in <code>tokens</code> with both instance variables (must be in the context of a loop)    | <ul style="list-style-type: none"> <li>access elements of <code>tokens</code> as if from an <code>ArrayList</code> (e.g., <code>tokens.get(i)</code>)</li> </ul>                                                                                                            | <ul style="list-style-type: none"> <li>use <code>==</code> for string comparison</li> <li>treat <code>tokens</code> as a single string</li> </ul>                                                                      |
| <b>+1</b>                                      | Adds delimiters into <code>ArrayList</code> in original order                                              | <ul style="list-style-type: none"> <li>add a delimiter by accessing <code>tokens</code> incorrectly (e.g., <code>tokens.get(i)</code>)</li> </ul>                                                                                                                           | <ul style="list-style-type: none"> <li>add a token that is not a delimiter</li> <li>do not maintain the original delimiter order</li> </ul>                                                                            |
| <b>Part (b)</b> <code>isBalanced</code>        |                                                                                                            | <b>5 points</b>                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                        |
| Points                                         | Rubric Criteria                                                                                            | Responses earn the point even if they...                                                                                                                                                                                                                                    | Responses will not earn the point if they...                                                                                                                                                                           |
| <b>+1</b>                                      | Initializes accumulator(s)                                                                                 | <ul style="list-style-type: none"> <li>initialize inside the loop</li> </ul>                                                                                                                                                                                                | <ul style="list-style-type: none"> <li>initialize an accumulator variable but don't update it</li> </ul>                                                                                                               |
| <b>+1</b>                                      | Accesses all elements in <code>ArrayList delimiters</code> (no bounds errors)                              | <ul style="list-style-type: none"> <li>return incorrectly inside the loop</li> </ul>                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>access elements of <code>delimiters</code> as if from an array (e.g., <code>delimiters[i]</code>)</li> </ul>                                                                    |
| <b>+1</b>                                      | Compares strings in <code>delimiters</code> with instance variables and updates accumulator(s) accordingly | <ul style="list-style-type: none"> <li>access elements of <code>delimiters</code> as if from an array (e.g., <code>delimiters[i]</code>)</li> </ul>                                                                                                                         | <ul style="list-style-type: none"> <li>use <code>==</code> for string comparison</li> <li>adjust an accumulator without a guarding condition</li> </ul>                                                                |
| <b>+1</b>                                      | Identifies and returns appropriate <code>boolean</code> value to implement one rule                        | <ul style="list-style-type: none"> <li>check for more closing delimiters (inside a loop) and return <code>false</code></li> <li>return <code>true</code> if the number of open and close delimiters is the same, and <code>false</code> otherwise (after a loop)</li> </ul> |                                                                                                                                                                                                                        |
| <b>+1</b>                                      | Identifies and returns appropriate <code>boolean</code> values for all cases                               | <ul style="list-style-type: none"> <li>have correct logic with the exception of a loop bounds error, accessing elements as if from an array, or using <code>==</code> for string comparison</li> </ul>                                                                      | <ul style="list-style-type: none"> <li>initialize accumulator inside a loop</li> <li>fail to check for more closing delimiters inside a loop</li> </ul>                                                                |

## 2019 SCORING GUIDELINES

## Question 3: Delimiters

## Part (a)

```
public ArrayList<String> getDelimitersList(String[] tokens)
{
    ArrayList<String> d = new ArrayList<String>();
    for (String str : tokens)
    {
        if (str.equals(openDel) || str.equals(closeDel))
        {
            d.add(str);
        }
    }
    return d;
}
```

## Part (b)

```
public boolean isBalanced(ArrayList<String> delimiters)
{
    int openCount = 0;
    int closeCount = 0;

    for (String str : delimiters)
    {
        if (str.equals(openDel))
        {
            openCount++;
        }
        else
        {
            closeCount++;
        }

        if (closeCount > openCount)
        {
            return false;
        }
    }

    if (openCount == closeCount)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.

## 2019 SCORING GUIDELINES

## Question 4: Light Board

|                 |            |                 |
|-----------------|------------|-----------------|
| <b>Part (a)</b> | LightBoard | <b>4 points</b> |
|-----------------|------------|-----------------|

**Intent:** Define implementation of a constructor that initializes a 2D array of lights

- +1** Creates a new `boolean[numRows][numCols]` and assigns to instance variable `lights`
- +1** Accesses all elements in the created 2D array (*no bounds errors*)
- +1** Computes the 40% probability
- +1** Sets all values of 2D array based on computed probability

|                 |               |                 |
|-----------------|---------------|-----------------|
| <b>Part (b)</b> | evaluateLight | <b>5 points</b> |
|-----------------|---------------|-----------------|

**Intent:** Evaluate the status of a light in a 2D array of lights

- +1** Accesses an element of `lights` as a `boolean` value in an expression
- +1** Traverses specified `col` of a 2D array (*no bounds errors*)
- +1** Counts the number of `true` values in the traversal
- +1** Performs an even calculation and a multiple of three calculation
- +1** Returns `true` or `false` according to all three rules

## Question-Specific Penalties

- 1** (z) Constructor returns a value
- 1** (y) Destruction of persistent data

## 2019 SCORING GUIDELINES

## Question 4: Scoring Notes

| Part (a) <code>LightBoard</code>    |                                                                                                           | 4 points                                                                                                                                                      |                                                                                                                                                                                                                            |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Points                              | Rubric Criteria                                                                                           | Responses earn the point even if they...                                                                                                                      | Responses will not earn the point if they...                                                                                                                                                                               |
| +1                                  | Creates a new <code>boolean[numRows][numCols]</code> and assigns to instance variable <code>lights</code> |                                                                                                                                                               | <ul style="list-style-type: none"> <li>initialize a local variable that is never assigned to <code>lights</code></li> <li>omit the keyword <code>new</code></li> <li>use a type other than <code>boolean</code></li> </ul> |
| +1                                  | Accesses all elements in the created 2D array ( <i>no bounds errors</i> )                                 | <ul style="list-style-type: none"> <li>fail to create <code>lights</code> but assume <code>lights[numRows][numCols]</code></li> </ul>                         |                                                                                                                                                                                                                            |
| +1                                  | Computes the 40% probability                                                                              | <ul style="list-style-type: none"> <li>use <code>Math.random() &lt;= .4</code></li> </ul>                                                                     | <ul style="list-style-type: none"> <li>incorrectly cast to <code>int</code></li> </ul>                                                                                                                                     |
| +1                                  | Sets all values of 2D array based on computed probability                                                 | <ul style="list-style-type: none"> <li>only assign <code>true</code> values</li> </ul>                                                                        | <ul style="list-style-type: none"> <li>compute a single probability but use it multiple times</li> <li>reverse the sense of the comparison when assigning</li> </ul>                                                       |
| Part (b) <code>evaluateLight</code> |                                                                                                           | 5 points                                                                                                                                                      |                                                                                                                                                                                                                            |
| Points                              | Rubric Criteria                                                                                           | Responses earn the point even if they...                                                                                                                      | Responses will not earn the point if they...                                                                                                                                                                               |
| +1                                  | Accesses an element of <code>lights</code> as a <code>boolean</code> value in an expression               |                                                                                                                                                               | <ul style="list-style-type: none"> <li>access <code>lights</code> as a type other than <code>boolean</code></li> </ul>                                                                                                     |
| +1                                  | Traverses specified <code>col</code> of a 2D array ( <i>no bounds errors</i> )                            |                                                                                                                                                               |                                                                                                                                                                                                                            |
| +1                                  | Counts the number of <code>true</code> values in the traversal                                            | <ul style="list-style-type: none"> <li>access too many or too few items in a single column</li> <li>access a single row instead of a single column</li> </ul> | <ul style="list-style-type: none"> <li>count an item more than once</li> </ul>                                                                                                                                             |
| +1                                  | Performs an even calculation and a multiple of three calculation                                          |                                                                                                                                                               | <ul style="list-style-type: none"> <li>use <code>/</code> instead of <code>%</code></li> </ul>                                                                                                                             |
| +1                                  | Returns <code>true</code> or <code>false</code> according to all three rules                              | <ul style="list-style-type: none"> <li>have an incorrect column count but use the correct logic</li> </ul>                                                    | <ul style="list-style-type: none"> <li>fail to return a value in some case</li> <li>implement counting loop more than once with one loop that is incorrect</li> </ul>                                                      |

## 2019 SCORING GUIDELINES

## Question 4: Light Board

## Part (a)

```
public LightBoard(int numRows, int numCols)
{
    lights = new boolean[numRows][numCols];

    for (int r = 0; r < numRows; r++)
    {
        for (int c = 0; c < numCols; c++)
        {
            double rnd = Math.random();
            lights[r][c] = rnd < 0.4;
        }
    }
}
```

## Part (b)

```
public boolean evaluateLight(int row, int col)
{
    int numOn = 0;

    for (int r = 0; r < lights.length; r++)
    {
        if (lights[r][col])
        {
            numOn++;
        }
    }

    if (lights[row][col] && numOn % 2 == 0)
    {
        return false;
    }
    if (!lights[row][col] && numOn % 3 == 0)
    {
        return true;
    }
    return lights[row][col];
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.