1. | Expression                          | Value |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>43 % 15 / 3 + 15 / 2</td>
<td>11</td>
</tr>
<tr>
<td>6.2 * 5 / 10 + 3.5</td>
<td>6.6</td>
</tr>
<tr>
<td>6 * 2.5 / 4 + (2.3 + 2.7) / 4</td>
<td>5.0</td>
</tr>
<tr>
<td>&quot;18&quot; + 3 * 4 + (8 + 5)</td>
<td>&quot;181213&quot;</td>
</tr>
<tr>
<td>59 % 10 / (2 + 2) * 2.5 / 2</td>
<td>2.5</td>
</tr>
</tbody>
</table>

2. The program produces the following output:
   one fish, two fish
   blue red, fish fish
   red one, two fish
   red fish, blue fish

3. | Method Call                     | Output Produced |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ifElseMystery(3, 1);</td>
<td>13 1 2</td>
</tr>
<tr>
<td>ifElseMystery(6, 9);</td>
<td>16 3 5</td>
</tr>
<tr>
<td>ifElseMystery(5, -1);</td>
<td>5 9 5</td>
</tr>
<tr>
<td>ifElseMystery(1, 2);</td>
<td>1 12 4</td>
</tr>
</tbody>
</table>

4. | Method Call                     | Output Produced |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mystery(5, 7);</td>
<td>2 -2 -2</td>
</tr>
<tr>
<td>mystery(4, 20);</td>
<td>16 13 11 10 10</td>
</tr>
<tr>
<td>mystery(10, 40);</td>
<td>30 21 13 6 0 0</td>
</tr>
<tr>
<td>mystery(5, 15);</td>
<td>10 6 3 1 0 0</td>
</tr>
</tbody>
</table>

5. | a != 0                          | c % 2 == 0      | b > 0 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point A</td>
<td>sometimes</td>
<td>always</td>
</tr>
<tr>
<td>Point B</td>
<td>always</td>
<td>sometimes</td>
</tr>
<tr>
<td>Point C</td>
<td>always</td>
<td>never</td>
</tr>
<tr>
<td>Point D</td>
<td>sometimes</td>
<td>sometimes</td>
</tr>
<tr>
<td>Point E</td>
<td>never</td>
<td>sometimes</td>
</tr>
</tbody>
</table>
6. One possible solution:
   public static void testFairCoin(Scanner console) {
      int heads = 0;
      int total = 0;

      System.out.print("next flip? ");
      String flip = console.next();
      while (!flip.equals("done")) {
         if (flip.equals("heads")) {
            heads++;
         } total++;
      }
      System.out.print("next flip? ");
      flip = console.next();
   }
   double pct = 100.0 * heads / total;
   System.out.println("was heads " + pct + \\ "% of the time");
}

7. One possible solution:
   public static int noBigger(int max, Random rand) {
      System.out.println("Picking numbers from 1 to " + max);
      int num = rand.nextInt(max) + 1;
      int last = num;
      int count = 1;
      while (num <= last) {
         double prob = (double)num / max;
         System.out.println("Probability to continue: " + prob);
         last = num;
         num = rand.nextInt(max) + 1;
         System.out.println("Number: " + num);
         count++;
      }
      System.out.println("Streak ends");
      return count;
   }

8. One possible solution:
   public static void trackInvestment(Scanner console, double init, int years) {
      System.out.println("Starting with: $" + init);
      double curr = init;
      for (int i = 1; i <= years; i++) {
         System.out.print("This year's return? ");
         int rate = console.nextInt();
         curr *= (1 + rate / 100.0);
         System.out.println("After year " + i + ": $" + curr);
      }
      System.out.println("Total interest earned: $" + (curr - init));
   }
9. One possible solution:

```java
public static int longestStreak(String str) {
    int max = 1;
    int len = 1;
    char curr = str.charAt(0);

    for (int i = 1; i < str.length(); i++) {
        if (str.charAt(i) == curr) {
            len++;
            max = Math.max(max, len);
        } else {
            len = 1;
            curr = str.charAt(i);
        }
    }

    return max;
}
```