1. | Expression | Value |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(5 \times 6 - (4 + 3) \times 2 - 2 \times 3)</td>
<td>10</td>
</tr>
<tr>
<td>(208 / 20 / 4 + 12 / 10.0 + 0.4 \times 2)</td>
<td>4.0</td>
</tr>
<tr>
<td>(8 - 2 + &quot;8 - 2&quot; + 8 \times 2 + 8)</td>
<td>&quot;68 - 2168&quot;</td>
</tr>
<tr>
<td>(4 \times 5 % 6 + 297 % 10 + 4 % 8)</td>
<td>13</td>
</tr>
<tr>
<td>(13 / 2 \times 3.0 + 5.0 \times 3 / 2)</td>
<td>25.5</td>
</tr>
</tbody>
</table>

2. The program produces the following output:

```
godel wrote grace with turing
borg wrote borg with boole
alan wrote hopper with lovelace
boole wrote boole with hopper
```

3. | Method Call | Output Produced |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ifElseMystery(1, 8);</td>
<td>3 8</td>
</tr>
<tr>
<td>ifElseMystery(3, 5);</td>
<td>5 0</td>
</tr>
<tr>
<td>ifElseMystery(4, 5);</td>
<td>5 6</td>
</tr>
<tr>
<td>ifElseMystery(8, 6);</td>
<td>8 2</td>
</tr>
<tr>
<td>ifElseMystery(7, 7);</td>
<td>7 8</td>
</tr>
<tr>
<td>ifElseMystery(5, 7);</td>
<td>7 2</td>
</tr>
</tbody>
</table>
4. | Method Call | Output Produced |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mystery(4);</td>
<td>2 2</td>
</tr>
<tr>
<td>mystery(5);</td>
<td>1 5</td>
</tr>
<tr>
<td>mystery(24);</td>
<td>4 3</td>
</tr>
<tr>
<td>mystery(28);</td>
<td>3 7</td>
</tr>
</tbody>
</table>

5. | x > 2 | x < n | n \% x == 0 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
<tr>
<td>Point A</td>
<td>Never</td>
<td>Sometimes</td>
</tr>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
<tr>
<td>Point B</td>
<td>Sometimes</td>
<td>Always</td>
</tr>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
<tr>
<td>Point C</td>
<td>Never</td>
<td>Sometimes</td>
</tr>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
<tr>
<td>Point D</td>
<td>Always</td>
<td>Sometimes</td>
</tr>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
<tr>
<td>Point E</td>
<td>Sometimes</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>+-----</td>
<td>+--------+</td>
</tr>
</tbody>
</table>
public static void selfCheckout(Scanner console, String saleItem, double discount) {
    System.out.print("How many items? ");
    int items = console.nextInt();
    double total = 0;
    double savings = 0;
    for (int i = 0; i < items; i++) {
        System.out.print("Item? ");
        String item = console.next();
        System.out.print("Price? ");
        double price = console.nextDouble();
        if (item.equals(saleItem)) {
            total += price * (1 - discount);
            savings += price * discount;
        } else {
            total += price;
        }
    }
    System.out.println("Final total (after discount): \$" + total);
    double perc = discount * 100;
    System.out.println("The \% discount on \$ saved you \$!");
}

public static void selfCheckout(Scanner console, String saleItem, double discount) {
    System.out.print("How many items? ");
    int items = console.nextInt();
    double total = 0;
    double savings = 0;
    for (int i = 0; i < items; i++) {
        System.out.print("Item? ");
        String item = console.next();
        System.out.print("Price? ");
        double price = console.nextDouble();
        if (item.equals(saleItem)) {
            savings += price * discount;
        } else {
            total += price;
        }
    }
    System.out.println("Final total (after discount): \$ + (total - savings));
    System.out.println("The \% discount on \$ saved you \$!");
}
7. One possible solution appears below:

```java
public static int noBigger(int max) {
    System.out.println("Picking numbers from 1 - " + max);
    Random r = new Random();

    int roll = r.nextInt(max) + 1;
    int limit = max;
    int count = 0;
    while (roll <= limit) {
        count++;
        System.out.println("Number: " + roll);

        limit = roll;
        double chance = 1.0 * limit / max;
        System.out.println("Probability to continue: " + chance);
        roll = r.nextInt(max) + 1;
    }
    System.out.println("Number: " + roll + ", streak ends");
    return count;
}
```

8. One possible solution appears below:

```java
public static int filter(int num, int d) {
    int result = 0;
    int multiplier = 1;
    while (num > 0) {
        int dig = num % 10;
        if (dig != d) {
            result += multiplier * dig;
            multiplier *= 10;
        }
        num /= 10;
    }
    return result;
}
```