Midterm Review

Approach: Work & discuss sample questions in class

Problem 1: 2 minutes
What does it print?

```java
int x = 3;
int y = 7;
x = x + y + x;
y = x - y;
System.out.println(y);
```

Problem 1: Approach

What does it print?

```java
int x = 3;
int y = 7;
x = x + y + x;
y = x - y;
System.out.println(y);
```

Review: Let’s work some problems and discuss answers and approaches.
Problem 2: 3 minutes

```java
int x = 16;
int y = 5;
x = x / y;
y = y % x;
System.out.println("x=" + x);
System.out.println("y=" + y);
```

Problem 3: 5 minutes

```java
int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);
```

Problem 2

```java
int x = 16;
int y = 5;
x = x / y;
y = y % x;
System.out.println("x=" + x);
System.out.println("y=" + y);
```

Problem 3 – first iteration

```java
int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);
```
Problem 3 – before 2\textsuperscript{nd} iteration

int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);

<table>
<thead>
<tr>
<th>i</th>
<th>k</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

Problem 3 – continue loop?

int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);

<table>
<thead>
<tr>
<th>i</th>
<th>k</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

Problem 3 – 2\textsuperscript{nd} iteration

int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);

<table>
<thead>
<tr>
<th>i</th>
<th>k</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>13</td>
</tr>
</tbody>
</table>

Problem 3 – 2\textsuperscript{nd} iteration

int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);

<table>
<thead>
<tr>
<th>i</th>
<th>k</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>13</td>
</tr>
</tbody>
</table>
Problem 3 – continue loop?

```
int k = 3;
int sum = 0;
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);
```

<table>
<thead>
<tr>
<th>i</th>
<th>k</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>13</td>
</tr>
</tbody>
</table>

Another way to work Problem 3

```
int k = 3;  // k = 3 (never changes)
int sum = 0;  // i = 5, 8, 11...
for (int i=5; i < 9; i+=k) {
    sum += i;
}
System.out.println(sum);
```

Problem 4: 7 minutes

```java
/**
 * Absolute value of difference.
 * @param x Any integer
 * @param y Any integer
 * @return | x - y |
 */
static int absdiff(
) {
}
```
One solution to Problem 4

static int absdiff(int x, int y) {
    if (x > y) {
        return x - y;
    } else {
        return y - x;
    }
}

And yes, indentation counts!

Another solution to Problem 4

static int absdiff2(int x, int y) {
    int result;
    result = x - y;
    if (result < 0) {
        result = 0 - result;
    }
    return result;
}

Problem 5: Area of rectangle

/** Computer area of rectangle with corners (x1,y1), (x2,y2).
 * (x1,y1) may be above or below, left or right of (x2,y2).
 * Areas are always positive. Zero means the two corners do
 * not define a rectangle, because the points are colinear.
 * All arguments are doubles.
 * @param x1 x coordinate of corner 1
 * @param y1 y coordinate of corner 1
 * @param x2 x coordinate of corner 2
 * @param y2 y coordinate of corner 2
 * @param area Area of rectangle defined by (x1,y1), (x2,y2),
 * or zero if (x1,y1) and (x2,y2) do not define a rectangle.
 */
static double rectArea( ) { // fill it in
}

A solution to Problem 5

static double rectArea( double x1, double y1,
                        double x2, double y2) {
    double area = (x2 - x1) * (y2 - y1);
    if (area < 0.0) {
        area = 0.0 - area;
    }
    return area;
}
One more ...

Write a method that prints
1
1 2
1 2 3
1 2 3 4
... (etc. up to) ...
1 2 3 4 5 6 7 8 9 10