

2. Parameters Simulation. 15 points. Consider the following program.

```
public class ParameterQuestion{

    public static void main(String[] args){
        int a = 2;
        int b = 3;
        int c = 2;
        int x = a + b;
        int y = a + 1;

        show(3, 2);
        show(x, y);
        show(c, c);
        show(a, b);

        System.out.println(a + " " + b);
    }

    public static void show(int a, int b){
        int c = a + b;
        a = b * 2;
        System.out.println(a + " " + b + " " + c);
    }
}
```

List below the output produced by this program when it is run.

3. Loops Simulation. 15 points. Consider the following method:

```
public static int loop(int x){  
    int total = 1;  
    for(int i = 1; i <= x; i++){  
        total = total * 2;  
    }  
    return total;  
}
```

For each method call below what value is returned?

Method Call	Value Returned
<code>loop(-1);</code>	_____
<code>loop(0);</code>	_____
<code>loop(1);</code>	_____
<code>loop(2);</code>	_____
<code>loop(3);</code>	_____

4. Simulation. 15 points. Sketch the drawing panel window that is produced when the following program when is run. Do not worry about the title bar.

```
import java.awt.*;

public class Draw{

    public static final int SIZE = 600;

    public static void main(String[] args){
        DrawingPanel p = new DrawingPanel(SIZE, SIZE);
        Graphics g = p.getGraphics();
        int increment = SIZE / 3;
        int x = 0;
        int y = 0;

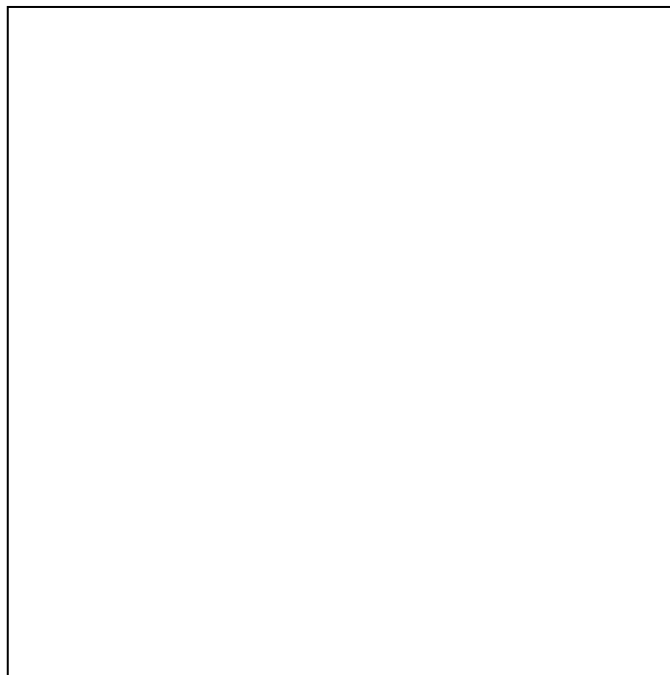
        // parameters on drawOval are x coordinate, y coordinate,
        // width, and height

        for(int row = 1; row <= 3; row++){
            x = 0;
            for(int column = 1; column <= 3; column++){
                g.drawOval(x, y, increment, increment);
                x = x + increment;
            }
            y = y + increment;
        }

        // parameters on drawLine are the x and y coordinates of the
        // two endpoints of the line: x1, y1, x2, y2

        g.drawLine(0, 0, SIZE, SIZE);
        g.drawLine(SIZE, 0, 0, SIZE);
    }
}
```

Sketch your
DrawingPanel
produced by the
program in the
box to the right.



5. Loops Simulation. 15 points. Consider the following program.

```
public class NestedLoops{

    public static final int SIZE = 5;

    public static void main(String[] args){
        String dot = ".";

        for(int i = 1; i <= SIZE; i++){

            for(int j = i; j <= SIZE; j++){
                System.out.print(dot);
            }

            for(int k = i; k <= SIZE; k++){
                System.out.print(i);
            }

            System.out.println();
        }
    }
}
```

List below the output produced by this program when it is run.

6. Programming. 15 points. In the imaginary game "Ring the Gack" players throw rings on the antlers of a "Gack", an imaginary creature from Dr. Seuss. Players throw red, blue, and yellow rings onto the antlers. They get 10 points for every red ring that lands on the antlers, 5 points for every blue ring that lands on the antlers, and 1 point for every yellow ring that lands on the antlers.

Write a method that given the number of red, blue, and yellow rings a player successfully threw onto the antlers returns the players score.

For example if a player had 3 red rings, 5 blue rings, and 2 yellow rings that landed on the antlers their score would be $30 + 25 + 2 = 57$.

You may assume the number of rings for each color given to the method will always greater than or equal to 0.

Complete you method below. You do not have to write an entire program, just the method described.

7. Programming. 15 points. Write a method that prints out a String with a specified number of characters per line. The method is given the String and the number of characters to print per line and then prints out that many characters per line, until every character in the String has been printed out. **You may assume that the number of characters in the given String is always a multiple of the given number of characters to print per line.**

For example if the String is "Isabelle" and the number of characters per line is 4, the output to the screen would be:

```
Isab  
elle
```

In this example the String "Isabelle" has 8 characters so the only valid characters per line would be 1, 2, 4, or 8.

Note all characters are treated the same. For example if the given String was "The Texas Longhorns!" and the number of characters per line was 5 the output would be:

```
The T  
exas  
Longh  
orns!
```

Note, the second line looks like it has 4 characters, but it actually has 5; the last one is a space. In this example the String "The Texas Longhorns!" has 20 characters so the only valid characters per line would be 1, 2, 4, 5, 10, or 20.

You may assume that the number of characters in the given String is always a multiple of the given number of characters to print per line. You will need to use the `length()` and `charAt()` methods from the String class. Recall the `charAt` method has one parameter, the index of the character to get from the String. The index of the first character is 0.

Complete your method below or on the next page. You do not have to write an entire program, just the method described.

There is more room on the next page if you need it.