1. Expressions. 2 points each, 18 points total. For each Java expression in the left hand column, indicate its value in the right hand column. **Be sure to show a constant of the appropriate type. For example, 7.0 rather than 7 for a double and "7" instead of 7 for a String.**

A. $2 + 3 \times 2$

B. $2.5 + 10 / 3$

C. $17 / 5$

D. $1 + 2 + "UT"$

E. $5 \% 12 \times 3 / 2$

F. "CS" + 3 + 7

G. $5.0 / 2 + 3$

H. $5 / 2 + 10 / 4$

I. $10 - 12 \times 5 / 2$
2. Parameters Simulation. 14 points. Consider the following program.

```java
public class ParameterQuestion{

    public static void main(String[] args){
        int x = 2;
        int y = 5;
        System.out.println( one(x, 10, y) );
        two(x, y);
        System.out.println( x + " " + y );
        int a = 3;
        int b = 2;
        two(b, a);
        System.out.println( a + " " + b );
        x = 3;
        y = four(x, x);
        System.out.println( y );
    }

    public static int one(int a, int b, int c){
        return Math.max( a, Math.max(b, c) );
    }

    public static void two(int a, int b){
        a++;
        b--;
        System.out.println( a + " " + b );
    }

    public static int three(int a){
        int x = a * 2;
        return x;
    }

    public static int four(int a, int b){
        a = a / 2;
        b = three(a) + three(b);
        System.out.println( a + " " + b );
        return a;
    }
}
```

List below the output produced by this program when it is run.
3. Loops Simulation. 8 points. Consider the following method:

```java
public static int loop(int val){
    int result = 0;
    for(int i = 1; i <= val; i++){
        result += i * 2;
    }
    return result;
}
```

For each method call below what value is returned?

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop(0);</td>
<td></td>
</tr>
<tr>
<td>loop(1);</td>
<td></td>
</tr>
<tr>
<td>loop(2);</td>
<td></td>
</tr>
<tr>
<td>loop(4);</td>
<td></td>
</tr>
</tbody>
</table>
import java.awt.*;

public class Draw{
    public static final int SIZE = 400;
    public static final int NUM_LINES = 4;

    public static void main(String[] args){
        DrawingPanel p = new DrawingPanel(SIZE, SIZE);
        Graphics g = p.getGraphics();
        int inc = SIZE / NUM_LINES;
        int end = inc;

        // parameters on drawLine are the x and y coordinates of the
        // two endpoints of the line: x1, y1, x2, y2
        for(int i = 1; i <= (NUM_LINES - 1); i++){
            g.drawLine(end, 0, end, SIZE);
            g.drawLine(0, end, SIZE, end);
            end = end + inc;
        }

        // parameters on drawOval are x coordinate, y coordinate,
        // width, and height
        g.drawOval(0, 0, SIZE, SIZE);
        g.drawOval(inc, inc, (inc * 2), (inc * 2));
    }
}

Sketch your
DrawingPanel
produced by the
program in the
box to the right.
Recall the origin
is at the top left
corner and y
increases as you
go down.
5. Programming. 10 points. Write a method that given the upper left and lower right coordinates of a rectangle on a Cartesian plane, returns the area of that rectangle.

Examples of results of method:

getArea(0.0, 4.0, 5.0, 0.5) returns 17.5
getArea(0.0, 4.0, 1.0, 0.0) returns 4.0
getArea(-2.5, 4.0, 2.5, -4.0) returns 40.0

Complete the method below:

// x1 and y1 indicate the upper left corner of the rectangle
// x2 and y2 indicate the lower right corner of the rectangle
public static double getArea(double x1, double y1,
                             double x2, double y2){

1 Sickle is worth 29 Knuts
1 Galleon is worth 17 Sickles
Based on these units 1 Galleon is worth 29 * 17 = 493 Knuts.

Write a method that given a number of Knuts prints out the equivalent number of Galleons, Sickles, and Knuts. The method prints out what the equivalent is in the fewest number of coins. (Galleons, Sickles, and Knuts are abbreviated to G, S, and K.)

Here are some examples of expected output:

showMoney(493) results in the following output:
493 Knuts is 1 G, 0 S, 0 K

showMoney(16) results in the following output:
16 Knuts is 0 G, 0 S, 16 K

showMoney(20) results in the following output:
20 Knuts is 0 G, 1 S, 3 K

showMoney(1100) results in the following output:
20 Knuts is 2 G, 6 S, 12 K

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

// knuts is the number of knut coins.
// knuts will be greater than or equal to 0
public static void showMoney(int knuts){
7. Programming. 25 points.

The harmonic series is the infinite series shown below:

\[ 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots \quad \text{also written as } \sum_{i=1}^{\infty} \frac{1}{i} \]

Write a method that given an int num, displays the first num terms in the harmonic series and shows the sum of those first num terms.

Examples of output:

showTerms(1) results in the following output:
1 = 1.0

showTerms(2) results in the following output:
1 + 1/2 = 1.5

showTerms(3) results in the following output:
1 + 1/2 + 1/3 = 1.8333333333333333

showTerms(8) results in the following output:
1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 = 2.7178571428571425

Note, there is one space on either side of each plus sign and the equals sign. There are not spaces between the numbers and the divide symbol.

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

// num is greater than or equal to 1
public static void showTerms(int num) {

