

Your Name

KEY AND CRITERIA

Problem Number	Topic	Points Possible	Points Off
1	Expressions	11	
2	Loop Evaluation	6	
3	Method Tracing	10	
4	Method Tracing	3	
5	Writing Methods	5	
6	Tracing Graphics	5	
7	Programming	10	
8	Using Graphics	10	
TOTAL POINTS OFF:			
SCORE OUT OF 60:			

Grading Acronyms:

NAP = No Answer Provided

LE = Logic Error

GCE = Gross Conceptual Error, based on answer did not understand question.

OBOE = Off By One Error

BOD = Benefit Of the Doubt

GACK = poor style or approach, but no points off

1. Expressions. 1 point each, 11 points total. For each Java expression in the left hand column, indicate its value in the right hand column.

You must show a value of the appropriate type. For example, 7.0 rather than 7 for a double and "7" instead of 7 for a String. Answers that do not indicate the data type correctly are wrong.

Wrong if missing quotes for Strings or decimal for doubles

A. `12 + "CS" + "\" + 1 + "E"` **"12CS"1E"**

B. `82617 / 1000 % 10 + 3 * 10` **32**

C. `1.5 + 6 / 4 + 2.5` **5.0**

D. `117 % 100 + 2 * 100 * 3 % 50` **17**

E. `29 / 4 / 2.0 + 18 / 5 + 1.5` **8.0**

F. `"UT" + "POP" + 3 * 5` **"UTPOP15"**

G. `3 * 2 + "C++" + 2 + 1` **"6C++21"**

H. `3 + 4 / (((double) 2) / 20)` **43.0**

I. `1.5 * 3 / 2 + 2 * 3` **8.25**

J. The three Math methods(ceil, floor, and abs) in the next expression all return doubles.

`Math.ceil(1.02) + Math.floor(2.4) + Math.abs(-1.5)` **5.5**

K. `((int) 15.6) * 3 + 2 * 3` **51**

2. Loops Simulation. 6 points total, 2 points each. Consider the following method:

```
public static void loop(int m, int p) {  
    final int LIMIT = m * p;  
    int temp = 0;  
    for(int i = 1; i <= LIMIT; i++) {  
        temp++;  
        temp += m;  
        p += 2;  
    }  
    System.out.print(m + " " + p + " " + temp);  
}
```

What is printed out by the following method calls?

-1 for each wrong number in order up to -2 per part

Method Call

Output to screen

`loop(-1, 3);`

-1 3 0

`loop(1, 4);`

1 12 8

`loop(-2, -2);`

-2 6 -4

3. Method Tracing and Parameters Simulation. (10 points) Consider the following program:

```
public class ParameterQuestion {

    public static void main(String[] args) {
        int a = 3;
        int b = first(a);
        System.out.println("A: " + a + " " + b);

        int x = 2;
        int y = 3;
        int z = second(x, y);
        System.out.println("B: " +x+ " " +y+ " " + z);

        a = third(-2, 3);
        System.out.println("C: " + a);

        a = 4;
        b = a / 2;
        int c = fourth(a, b);
        System.out.println("D: " +a+ " " +b+ " " + c);
    }

    public static int first(int a) {
        a += 2;
        return a * 2;
    }

    public static int second(int a, int b) {
        a = first(a);
        b = first(b);
        return a + b;
    }

    public static int third(int a, int b) {
        int temp = a;
        a = b;
        b = temp;
        System.out.println("E: " + a + " " + b);
        return a;
    }

    public static int fourth(int a, int b) {
        int y = second(b, a);
        return y - b;
    }
}
```

List output of program here:

A: 3 10
B: 2 3 18
E: 3 -2
C: 3
D: 4 2 18

**-1 per wrong number
per line up to -2
per line**

**-1 if wrong order on
letters**

In the box to the above list the output produced when the program runs.

List the output in the order it will appear on the screen when the program runs.

4. Method Tracing. 3 points. What is output when the following program is run?

```
public class MethodTrace {  
  
    public static void main(String[] args) {  
  
        int x = 5;  
        int y = 10;  
        System.out.println( toy(x) + " " + x + " " + toy(y));  
    }  
  
    public static int toy(int x) {  
        x = x * x;  
        System.out.print(x);  
        return x;  
    }  
}
```

-1 for each value out of order or wrong.

OUTPUT: **2510025 5 100**

5. (5 points) Write a method that calculates and returns a student's GPA given the number of A's, B's, C's, D's, and F's the student has earned in classes. The number of each letter grade earned (A, B, C, D, and F) are sent as parameters to the method. Assume every class the student takes is worth 3 credits. A's are worth 4 grade points, B's are worth 3 grade points, C's are worth 2 grade points, D's are worth 1 grade point, and F's are worth 0 grade points.

For example, given a student with 3 A's, 4 B's, 1 C, 0 D's, and 2 F's the method returns a GPA of 2.6.

```
public static double getGPA(int a, int b, int c,  
                             int d, int f) {  
    int totalClasses = a + b + c + d + f;  
    int points = a * 4 + b * 3 + c * 2 + d;  
    return ((double) points) / totalClasses;  
}
```

header: 1 point

calculate total classes: 1 point

calculate grade points: 1 point

return value: 1 point

floating point division: 1 point

hard coding number of each grade: - 4

6. Tracing Graphics Programs. 5 points. Sketch the drawing panel window that is produced when the following program when is run. Do not draw the title bar.

```
import java.awt.*;

public class Draw {

    public static final int SIZE = 400;

    public static void main(String[] args) {
        DrawingPanel p = new DrawingPanel(SIZE, SIZE);
        Graphics g = p.getGraphics();
        int part = SIZE / 4;

        // parameters on drawRect and fillOval are x, y, width, height

        g.drawRect(part, 0, part, SIZE);
        g.drawRect(0, part, SIZE, part);

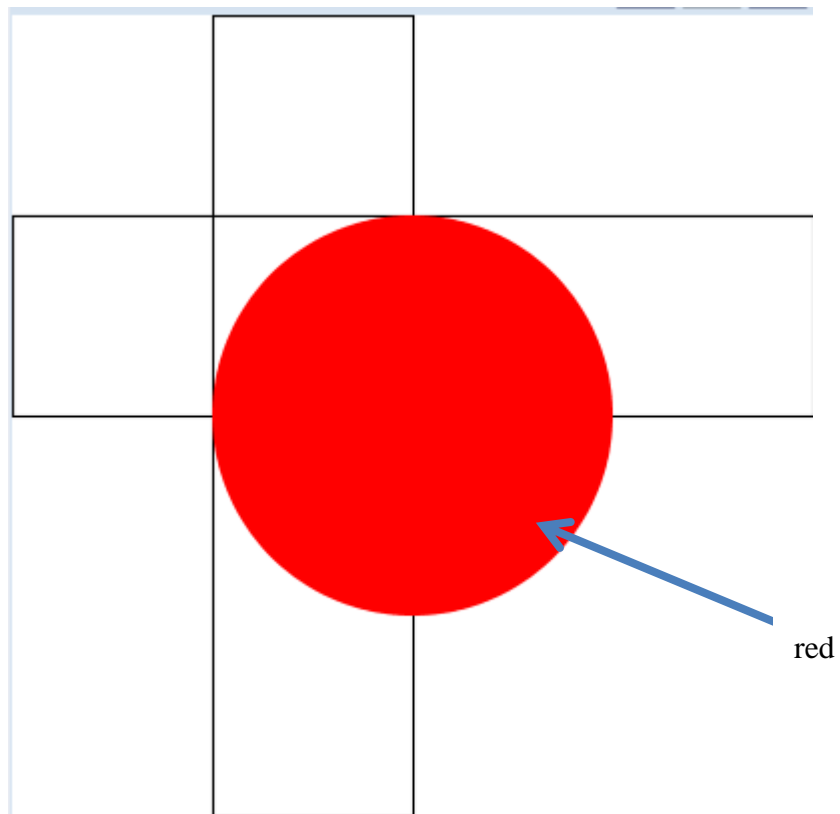
        g.setColor(Color.RED);
        g.fillOval(part, part, SIZE / 2, SIZE / 2);

    }
}
```

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

Indicate the color
with a label if it
is not black.

Recall the origin
is at the top left
corner and y
increases down
the panel, not up.



2 rectangles: 1 point

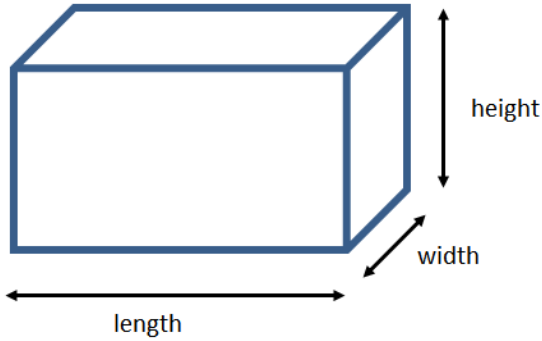
1 circle: 1 point

circle filled and labeled red: 1 point

rectangles approximately correct position and size: 1 point

circle approximately correct position and size: 1 point

7. Programming. 10 points. Write a method that prints out the volume of a box for various values of length, width, and height.



The method shall have 3 parameters for the maximum length, width, and height. Print out all volumes of the box as the length, width, and height, vary from 1 to the maximum value. For example, given a maximum length of 2, a maximum width of 3, and a maximum height of 2 your method shall produce the following output:

```
LEN: 1 WID: 1 HT: 1 VOL: 1
LEN: 1 WID: 1 HT: 2 VOL: 2
LEN: 1 WID: 2 HT: 1 VOL: 2
LEN: 1 WID: 2 HT: 2 VOL: 4
LEN: 1 WID: 3 HT: 1 VOL: 3
LEN: 1 WID: 3 HT: 2 VOL: 6
LEN: 2 WID: 1 HT: 1 VOL: 2
LEN: 2 WID: 1 HT: 2 VOL: 4
LEN: 2 WID: 2 HT: 1 VOL: 4
LEN: 2 WID: 2 HT: 2 VOL: 8
LEN: 2 WID: 3 HT: 1 VOL: 6
LEN: 2 WID: 3 HT: 2 VOL: 12
```

```
public static void printVolumes(int ml, int mw, int mh) {
    for(int len = 1; len <= ml; len++) {
        for(int wid = 1; wid <= mw; wid++) {
            for(int ht = 1; ht <= mh; ht++) {
                int volume = len * wid * ht;
                System.out.println("LEN: " + len +
                                   " WID: " + wid + " HT: "
                                   + ht + " VOL: " + volume);
            }
        }
    }
}
```

header: 1 point

parameters: 1 point

3 loops: 1 points

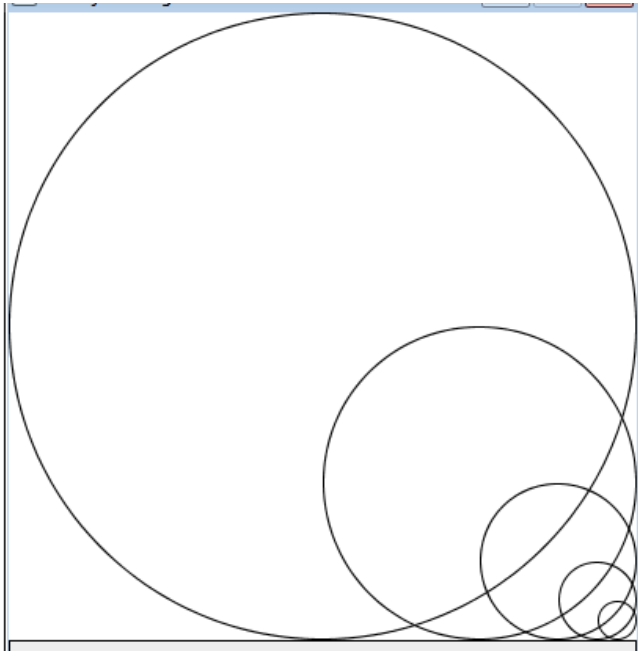
loops nested correctly: 3 points

correct limits on loops, use of parameters: 1 point

calculate volume correctly inside nested loop: 2 point (must be in triply nested loop

println correct inside nested loop: 1 points

8. Graphics Programming. 10 Points. Complete a method to produce the following output. The `Graphics` object for the `DrawingPanel` is sent to the method as a parameter as well as the size of the `DrawingPanel`. Assume the `DrawingPanel` is square with length and width equal to the `size` parameter.



A given oval has the same width and height. (They are circles.)

The width and height of each successive circle is half of the previous circle.

Each successive circle is drawn in the lower right quadrant of the previous circle.

The method always draws 5 circles.

Assume the color of the `Graphics` object has already been set to `Color.BLACK`.

```
public static void drawCircles(Graphics g, int size) {  
    int coord = 0;  
    int width = size;  
    for(int i = 1; i <= 5; i++) {  
        g.drawOval(coord, coord, width, width);  
        width = width / 2;  
        coord += width;  
    }  
}
```

header with parameters correct: 2 points

call `drawOval` correct: 2 points

draws 5 circles total: 2 points (-1 on width / height not necessary)

draw circles in correct positions: 2 points

draw circles of correct sizes: 2 points

create new `DrawingPanel`: -1

hard coding the 5 ovals is okay if math correct.