CS 312 – Exam 1 – Fall 2022 - SOLUTIONS

Expressions and Statements - 3 point each

A) 30
B) 38
C) 5.5
D) 7.25
E) “50UT”
F) “802”
G) -4
H) 18.0
I) -27
J) 3
K) False
L) False
~~M) 27~~ removed from the test
~~N) K. Johnson~~ removed from the test

Code Tracing - 7 point each

1. D
2. A
3. D
4. C
5. A
6. B
7. D
8. D
9. A
10. B
11. D
12. C

Program - Roll Dice - 44 points

```
1     public static void main(String[] args) {
2         Scanner console = new Scanner(System.in);
3         System.out.print ("How many times do you want to roll the die? ");
4         int numRolls = console.nextInt();
5         for (int i=2; i<=12; i++) {
6             System.out.print("The value " + i + " was rolled " + roll(i, numRolls) + " times.");
7         }
8     }
9     private static int roll (int value, int numRolls) {
10        int count = 0;
11        int die1, die2;
12        for (int i=0; i<numRolls; i++) {
13            die1 = (int)(Math.random() * 6 + 1);
14            die2 = (int)(Math.random() * 6 + 1);
15            if (die1 + die2 == value) count++;
16        }
17       return (count);
18    }
```

Line #		Points
MAIN method		
1	Main header	N/A
2	scanner	+1 attempt to make a scanner +1 made a scanner correctly
3	prompt	+2 prompted the user to get roll count
4	numRolls	+1 created numRolls variable +1 used console.nextInt()
5	Loop 2-12	+1 good for loop +1 start at 2, end at 12
6	Method call/string	+1 correct method call +1 correct string
ROLL method		
9	Method declsration	+2 int return value +2 send role outcome as int parameter +2 send int numRolls as parameter
10	Count variable	+1 declare count +1 initialize count
11	Die variables	+ 1 declare die1 (in or out of for loop) +1 declare die2
12	Loop to roll die	+2 initialize correctly +2 test correctly +2 increment
13 and 14	2 separate rolls	+2 use Math.random +2 correct range +2 call random twice

15	Compare to current value	+3 test die tolal vs. value +3 increment count
17	Return count	+3 have a return +3 return count

-2 Extra code that is no good
-2 use code you shouldn't have

```
 public static void main(String[] args) {
     Scanner console = new Scanner(System.in);
     System.out.print ("How many times do you want to roll the die? ");
     int numRolls = console.nextInt();
     for (int i=2; i<=12; i++) {
         System.out.println("The value " + i + " was rolled " + roll(i, numRolls) + " times.")
     }
 }


 private static int roll (int value, int numRolls) {
     int count = 0;
     int die1, die2;
     for (int i=0; i<=numRolls; i++) {
         die1 = (int)(Math.random() * 6 + 1);
         die2 = (int)(Math.random() * 6 + 1);
         if (die1 + die2 == value) count++;
     }
    return (count);
}
```

Program - Graphics - 60 points total

Part 1 - Annotate Graphic - 10/60 points

8+ annotations exist

Part 2 - Loop Tables - 15/60 points

Each column in grey counts for 1 point when all the numbers are correct.

Panel height	Panel width	Circle diameter	X offset (for whole figure)	Y offset (for whole figure)	Number of Circles (# top to bottom or left to right)
400	400	20	100	100	10
SIZE	SIZE	SIZE/20	SIZE/4	SIZE/4	SIZE/20

Create a Loop Table for the top half that draws the blue circles.

Row (i)	**numSpots**	**Number of Empty Spots**	**Number of Circles**	**Y offset (same for all circles on a given line)**	**All the js**	**X offset at each j**
1	5	4	1	100	1	180
2	5	3	2	120	1 2	180 160
3	5	2	3	140	1 2 3	180 160 140
4	5	1	4	160	1 2 3 4	180 160 140 120
5	5	0	5	180	1 2 3 4 5	180 160 140 120 100
MATH		numSpots-i	i	SIZE/4 + (i-1) * diameter	Start at 1 End at i	SIZE/4 + (numSpots - j) * diameter

Create your own Loop Table for the bottom half that draws the red circles.

Row (i)	**Num Spots**	**Number of Empty Spots**	**Number of Circles**	**Y offset (same for all circles on a given line)**	**All the js**	**X offset at each j**
1	5	0	5	200	1 2 3 4 5	200 220 240 260 280
2	5	1	4	220	1 2 3 4	200 220 240 260
3	5	2	3	240	1 2 3	200 220 240
4	5	3	2	260	1 2	200 220
5	5	4	1	280	1	200
Math		i-1	numSpots - i + 1	SIZE/2 + (i-1) * diameter	Start at 1 End at numSpots -i + 1	SIZE/2 + (j-1) * diameter

Part 3 - drawFigure method - 35 points

```
public static void drawFigure(Graphics gr) {
    int diameter = SIZE/20;
    int xOffset=0, yOffset=0;
    int numSpots = SIZE / 4 / diameter; // num spots per section
    gr.setColor(Color.BLUE);
    // top left
    for (int i=1; i<=numSpots; i++) {
        yOffset = SIZE/4 + diameter * (i - 1);
        for (int j=1; j<=i; j++) {
            xOffset = SIZE/4 + diameter * (numSpots - j);
            gr.fillOval(xOffset, yOffset, diameter, diameter);
        }
    }
    // bottom right
    gr.setColor(Color.RED);
    for (int i=1; i<=numSpots; i++) {
        yOffset = SIZE/2 + diameter * (i - 1);
        for (int j=1; j<=(numSpots-i+1); j++) {
            xOffset = SIZE/2 + diameter *(j-1);
            gr.fillOval(xOffset, yOffset, diameter, diameter);
        }
    }
} // end of drawFigure()
```

#	Item	Points	Description
a	Diameter formula	2	Correct formula (see table for formulas)
b	numSpots	1	Correct formula (formula or 5)
c	Blue	1	SetColor to blue for top half
d	Red	1	Set color to red for bottom half
TOP LEFT			
e	outside loop	3	executes correct number of times (1 - numSpots)
f	yOffset formula	3	Correct formula (multiple ones will work)
g	inside loop	3	Executes correct number of times (1 to i)
h	xOffset	3	Correct formula (multiple ones will work)
i	fillOval	3	Correct parameters - x, y, diameter and diameter

BOTTOM RIGHT			
j	outside loop	3	executes correct number of times (1 - numSpots)
k	yOffset formula	3	Correct formula
l	Inside loop	3	Executes correct number of times (1 to i)
m	xOffset formula	3	Correct formula
n	fillOval	3	Correct parameters - x, y, diameter and diameter