CS 312 – Exam 1 – Fall 2018

Name: _____ UTEID: _____

Circle your TA's Name: Aish Anthony Carla Daniel Dayanny

Fatima Kate Mallory Rebecca

Problem Number	Topic	Points Possible	Points Off
1	Expressions	15	
2	Code Tracing	20	
3	Syntax Errors	12	
4	Programming	8	
5	Programming	12	
6	Programming	18	
7	Graphics	15	
TOTAL POINTS OFF:			
SCORE OUT OF 100:			

Instructions:

1. Please turn off or silence your mobile devices.
2. You have 2 hours to complete the test.
3. Place your final answers on this test, not scratch paper. Answers must be in pencil.
4. You may not use a calculator.
5. When code is required, write Java code. You may use only features that we discussed up to topics 1-12, including those covered in the textbook for that material (Chapters 1-4).
6. You may write additional methods to provide structure and remove redundancy.
7. The exam proctors will not answer questions regarding the content of the exam. If you believe a question has an error or is ambiguous, state your assumption and answer based on your assumption.
8. If you finish early bring your exam and scratch paper to the proctor and show them your UTID.

1. Expressions: 1 point each, 15 points total. For each Java expression in the left hand column, indicate the result of the expression 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.

A. `5 + 3 * 7 / 2` _____

B. `142 / 10 + 17 / 34` _____

C. `7.0 / 2.0 + 7.5 / 2.5` _____

D. `37.5 / 10.0 + 5 / 10 + 7 / 2.0` _____

E. `4 - 2 + "4" + 2 + 2 * 4` _____

F. `3 * -4 + "DHF" + (6 - 10)` _____

G. `40 % 5 - 35 % 6 + 15 % 30` _____

H. `2.9 % 1.2 + 1635 % 5` _____

I. `"" + -3 + 5 + 45 / 10` _____

J. `(double) (16 / 5)` _____

K. `(int) (.98765 * 10)` _____

L. `(double) 10 / 4 + 15 / 2` _____

M. `Math.pow(Math.floor(2.95), Math.ceil(4.0))` _____

N. `Math.floor(-39.75 / 10.0)` _____

O. `15 / 4 / 3.0 - 18 / 5 + (15 / 10.0)` _____

2. Code tracing: 2 points each, 20 points total. Place you answer in the box to the right of the code. If the code results in a syntax error, answer **syntax error.** If the code results in a runtime error, answer **runtime error. For output show exactly what is output to the screen when the code is run.**

A. What is output by the following code when it is run?

```
int x1 = 3;
int y1 = x1 * 3;
double a1 = x1 / 2 + y1 / 2;
x1 = x1 + 2;
y1--;
a1 = a1 / 2;
System.out.print(x1 + " " + y1 + " " + a1);
```

B. What is output by the following code when it is run?

```
int x2 = 1;
int y2 = 10;
y2 *= 1 + x2 * (y2 - 12);
System.out.print(y2 - 10);
```

C. What is output by the following code when it is run?

```
String s3 = "x3";
int x3 = 4;
double a3 = 2;
s3 = x3 + 2 + s3 + a3 + 1;
System.out.print(s3);
```

D. How many asterisks does the following code print out?
Don't show the output. Simply state the number of asterisks that are printed out when the code runs

```
System.out.print("*");
for (int i = -1; i <= 6; i++) {
     System.out.print("*");
     System.out.print("*");
}
System.out.print("*");
```

E. How many asterisks does the following code print out?
Don't show the output. Simply state the number of asterisks that are printed out when the code runs

```
for (int i = 1; i <= 5; i++) {
     System.out.print("**");
     for (int j = 1; j <= 10; j++) {
          System.out.print("**");
          System.out.print("***");
     }
     System.out.print("*");
}
```

F. How many asterisks does the following code print out?
Don't show the output. Simply state the number of asterisks that are printed out when the code runs

```
for (int i = 0; i < 5; i++) {
     for (int j = 1; j <= i; j++) {
          System.out.print("*");
     }
     for (int j = 0; j < 3; j++) {
          System.out.print("*");
          for (int k = 0; k < 3; k++) {
               System.out.print("*");
          }
     }
}
```

G. What is output by the following code when it is run?

```
int x7 = 6;
int y7 = 3;
int z7 = y7 * 10 % (x7 - (y7 * 2));
System.out.print(z7);
```

H. What is output by the following code when it is run?

```
int x8 = 8;
int y8 = -5;
methodH(y8, x8);
methodH(x8, x8);
System.out.print(x8 + " " + y8);

public static void methodH(int x, int y) {
     x++;
     y--;
     System.out.print((x + y) + " ");
}
```

I. What is output by the following code when it is run?

```
int x9 = 5;
int y9 = 10;
x9 = methodI(x9, y9);
System.out.print(x9 + " " + y9);

public static int methodI(int x, int y) {
     x--;
     y *= 2;
     return x + y;
}
```

J. What is output by the following code when it is run?

```
int x = 2;
int y = 3;
System.out.print(methodJ(x, y) + x * y + methodJ(y, y));

public static int methodJ(int x, int y) {
     x += 2;
     y -= 2;
     System.out.print(x * y);
     return x + y;
}
```

3. Syntax Errors: 12 Points. ***Most*** of the following snippets of code contain syntax errors.
If the snippet contains a syntax error, circle the line of code with the syntax error AND explain the error in one sentence. You don't have to give exact output of the compiler, but you must explain why the error occurs.
If the snippet does not contain a syntax error, answer **No Error.**
Assume each snippet is contained in a distinct method.

A. Answer:

```
// A
System.out.println("Line 1"\nLine2");
System.out.println("Line 3");
System.out.println();
System.out.println("Line 5");
```

B. Answer:

```
// B
int x2 = 72;
int y2;
y2 = x2 / 10 + 3;
int x2 = y2 * 5;
y2++;
```

C. Answer:

```
// C
double Int = 8.5;
double a3 = Int / .02;
int x3 = (int) a3 + 225;
x3 += 5 / 10;
```

D. Answer:

```
// D
String s4 = "Java";
int x4 = 10;
int y4 = 3;
s4 = s4 + x4 - y4 * 5;
String s5 = s4 + s4 + s4;
```

E. Answer:

```
// E
int x5 = 10;
int y5 = 3;
double b5 = x5 / y5 + (x5 * 2) + (y5 * 3);
double a5 = b5 / 2.5;
```

F. Answer:

```
// F
STRING s6 = "5 + 4";
System.out.println(4 - 5 + s6 + "   10\3 = 3 in Java");
System.out.println(s6);
```

G. Answer:

```
// G
int result = 0;
for (int i = 4;i <= 25;i++) {
     int temp = i * 3;
     resulttt += temp;
}
System.out.println(result);
```

H. Answer:

```
// H
double total = 0.0;
for (int i = 0; i < 100; i++) {
     double temp = Math.sqrt(i * 1.0);
     total = temp + total;
}
System.out.println(total + " " + i);
```

I. Answer:

```
// I
double a7 = 3.7;
String s7 = a7 + (5 - 10);
```

J. Answer:

```
// J
public static int methodJ(int x, int y) {
     System.out.println(x);
     x++;
     y--;
     System.out.println(x + y);
}
```

K. Answer:

```
// K
String s9 = "**";
int y = s9.length() + 10;
methodK(s9, y);
System.out.println(s9 + " " + y);

public static void methodK(int x, String s) {
     System.out.print(x + " " + s);
}
```

L. Answer:

```
// L
int ck;
int garg = 15;
ck++;
garg -= Math.abs(100);
```

4. Programming: 8 points Write a static method, `getQuadrant`. The method header is

```
public static int getQuadrant(double x, double y)
```

Given the x and y coordinate return the quadrant the point specified by the parameters is in.

If the given point is on the origin or the x-axis or the y-axis return 0.

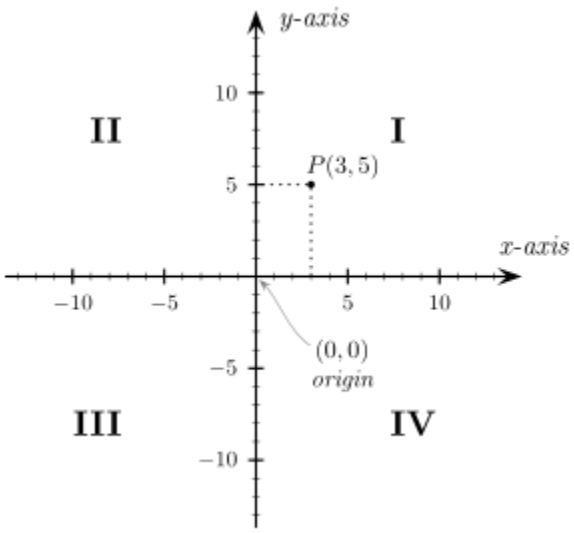


Image Source:
https://en.wikipedia.org/wiki/Quadrant_(plane_geometry)

Recall the Cartesian plan quadrants: quadrant 1 is the upper right quadrant, quadrant 2 is the upper left quadrant, quadrant 3 is the lower left quadrant, and quadrant 4 is the lower right quadrant.

Examples of calls to the method:

```
getQuadrant(-2.5, -3.0) -> returns 3
getQuadrant(-0.5, 0.0)  -> returns 0
getQuadrant(3.0, 5.0)   -> returns 1
```

Complete the method below:

```
public static int getQuadrant(double x, double y) {
```

5. Programming: 12 points - Complete the `printFigure` method. The method header is:

```
public static void printFigure(int n)
```

Examples of calls to `printFigure`:

If `n` = 1, the following is printed:
```
(0)
```

If `n` = 2, the following is printed:
```
(0)
22(2)
```

If `n` = 3, the following is printed:
```
(0)
22(2)
3333(4)
```

If `n` = 4, the following is printed:
```
(0)
22(2)
3333(4)
44444444(8)
```

If `n` = 5, the following is printed:
```
(0)
22(2)
3333(4)
44444444(8)
555555555555(12)
```

If `n` = 6, the following is printed:
```
(0)
22(2)
3333(4)
44444444(8)
555555555555(12)
6666666666666666(18)
```

If `n` = 7, the output would be the same as `n` = 6, except with this as the additional last line
```
777777777777777777777777(24)
```

The number in parenthesis at the end of each line is part of the expected output.

You may assume the parameter `n` is > 0.

Do not use conditionals (if statements) in your answer. Use the print and println methods, but no others.

Complete the `printFigure` method on the next page.

```
public static void printFigure(int n) {
```

6. Programming: 18 points. Write the `rollDie` method explained below. You must write the method header and the code for the method.

The method accepts two parameters:

- a `Scanner` object. Assume the `Scanner` object is already connected to `System.in`.
- an integer that represents a target value

The method prompts the users for the number of times to roll a die and the number of sides on the die. The sides of the die are numbered 1 to N where N is the number of sides. So a 4 side die has sides 1, 2, 3, and 4. The die is a fair die meaning when it is rolled each side has an equal probability of being the result.

- You may assume the user always enters integer values for the number of rolls and the number of sides.
- If the user enters an integer less than 1 for the number of rolls, make the number of rolls 10.
- If the user enters an integer less than 3 for the number of sides on the die, make the number of sides on the die 6.

The method then simulates rolling a die with the given number of sides the given number of times.

When the die has been rolled the appropriate number of times the method prints out the sum of all the rolls of the die and whether the sum exceeded the target value sent as a parameter. The sum of the rolls must be greater than the target to succeed.

Example output of the method being called two times. The users input is bolded:

```
Times to roll: 4
Sides on die: 5
rolled 2
rolled 4
rolled 1
rolled 4
Sum of rolls: 11
15 not exceeded

Times to roll: -2
Sides on die: 2
rolled 5
rolled 3
rolled 5
rolled 4
rolled 3
rolled 4
rolled 3
rolled 1
rolled 5
rolled 6
Sum of rolls: 39
25 exceeded
```

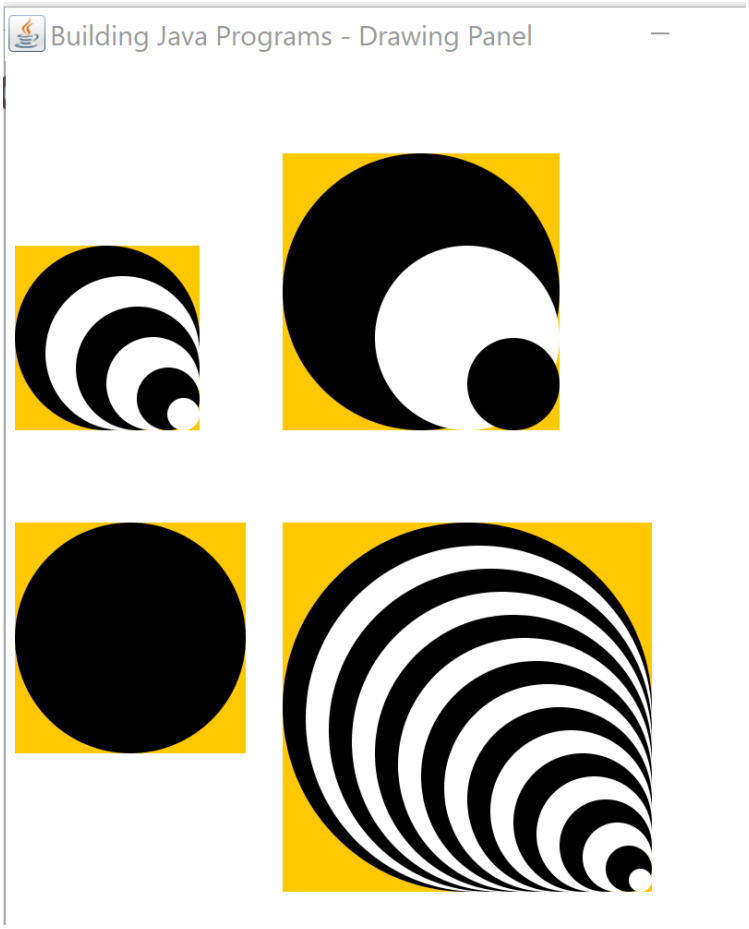
You may use the `Math.random()` method to generate random numbers, the Scanner nextInt() method, and the println and print methods. Do not use any other Java classes or methods.

Complete the `rollDie` method on this page, including the method header.

7. Graphics Programming: 15 points Complete a method, `drawFigure` that produces the following figure using Java 2D graphics. The parameters for the method are:

- The `Graphics` object for the `DrawingPanel`. You do not know what the current `Color` is.
- The x and y coordinates of the upper left corner of the figure.
- The size of the figure. The figure is square.
- The number of circles in the figure

The following image shows the results of various calls to `drawFigure`.



The background square is orange. The circles in the figure alternate between black and white.

The figure at the top left has a size of 210. The six circles in that figure have diameters of 210, 175, 140, 105, 70, and 35.

The figure at the top right has a size of 300. The three circles in that figure have diameters of 300, 200, and 100.

The diameter of the circles in the other figures follow the same pattern based on size of figure and number of circles.

Recall the following methods from the Graphics class:

```
fillRect(int x, int y, int width,
     int height)

fillOval(int x, int y, int width,
     int height)

setColor(Color c)
```

Recall the constants from the `Color` class: `Color.ORANGE`, `Color.BLACK`, and `Color.WHITE`.

The figures above are produced by the following calls to `drawFigure(Graphics g, int x, int y, int size, int numCircles)`:

```
drawFigure(g, 10, 200, 210, 6);    // top left
drawFigure(g, 300, 100, 300, 3);   // top right
drawFigure(g, 10, 500, 250, 1);    // lower left
drawFigure(g, 300, 500, 400, 16);  // lower right
```

Complete the method on the next page.

```
public static void drawFigure(Graphics g, int x, int y, int size,
                              int numCircles) {
```