CS 312 – Exam 1 – Fall 2017

Name: _____ UTEID: _____

Circle your TA's Name: Aish Anthony Bri Carla Chris

CK Dayanny Fatima Hailey Omer

Problem Number	Topic	Points Possible	Points Off
1	Expressions	20	
2	Code Tracing	18	
3	Method Trace	14	
4	Programming	12	
5	Programming	12	
6	Programming	12	
7	Graphics	12	
TOTAL POINTS OFF:			
SCORE OUT OF 100:			

Instructions:

1. Please turn off your cell phones.
2. There are 7 questions on this test.
3. You have 2 hours to complete the test.
4. Place your final answers on this test, not scratch paper.
5. You may not use a calculator.
6. 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).
7. Style is not evaluated when grading. You may add more methods if you wish.
8. The proctors will not answer questions. If you believe a question has an error or is ambiguous, state your assumption and answer based on your assumption.
9. If you finish early bring your exam and scratch paper to the proctor and show them your UTID.

1. Expressions: 1 point each, 20 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. `-1 * 5 - 8 * 5 - 15` _____

B. `25 / 8 + 12 / 7 + 21 / 6` _____

C. `2 + 15 * 10 - 6 / 10` _____

D. `22 / 10 * 7 % 3 * 10 / 2` _____

E. `13672 % 1000 / 100 + 16 % 4 * 3` _____

F. `3.5 % 3.0 + 2.5 * 3` _____

G. `7.0 / 2.0 + 13.0 / 10.0 - 1.0` _____

H. `5 * (3 + 2) / (6 - 8) + 4` _____

I. `7 / 2 + 1.5 + 10 / 4` _____

J. `15 / 20 + 20 / 15 + 8 / 10` _____

K. `6 / 10.0 + 6 / 10 + 25 / 100.0` _____

L. `3 * 2 + "*" + 2 * 4 + "!"` _____

M. `"CS" + 31.2 * 10 + "??"` _____

N. `3 - 7 + 1 + "PY" + 6 + -3` _____

O. `1.5 + 20 / 7 + 13 % 65 * 2 - .2` _____

P. `((int) (5.0 / 10)) + (double) 5 / 10` _____

Q. `(int) (.075 * 10 + 3 / 2)` _____

R. `"val" + (2 * 1.5) + (6 % 2)` _____

The `Math` methods `ceil` and `pow` return `doubles`.

S. `Math.ceil(-.99 * 10)` _____

T. `Math.pow(2.0, 3.0)` _____

2. Code tracing: 2 points each, 18 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.**

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

```
int x1 = 5;
int y1 = 2 * x1 + 2;
x1 = x1 * 2 + x1 * 3;
int z1 = x1 + y1 / 3 + x1;
z1 += 3;
System.out.print(x1 + " " + y1 + " " + z1);
```

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

```
int x2 = 5;
int y2 = 3;
x2 *= 2;
x2 = x2 / ((y2 - y2) * 3);
System.out.print(x2 + " " + y2);
```

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

```
int x3 = 5;
double a3 = x3 * 3 / 10;
a3 *= 3.0;
x3--;
System.out.print(x3 + " " + a3);
```

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

```
double a4 = 1.5;
int x4 = 10;
x4 = x4 + a4 * 2;
a4 = 2.5;
System.out.print(a4 + " " + x4);
```

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

```
int x5 = 3;
int y5 = -1 - x5;
x5 *= 2 + y5;
System.out.print(x5 + " " + y5);
```

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

```
String s6 = "ABC";
String t6 = "\"Q\"";
int x6 = 12;
s6 = x6 + t6;
t6 = "JJ";
System.out.println(s6 + " " + t6);
```

G. 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 <= 20; i++) {
     System.out.print("***");
}
```

H. 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 = -5; i < 6; i++) {
    System.out.print("*");
    for (int j = 0; j < 10; j++) {
         System.out.print("*");
    }
}
```

I. 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 < 10; i++) {
     for(int j = 0; j < 7; j++) {
          System.out.print("*");
     }
     for(int j = 1; j < 5; j++) {
          for (int k = 0; k < 10; k++) {
              System.out.print("*");
          }
     }
}
```

3. Method Tracing and Parameters Simulation: 2 points each, 14 points total.
For each part write what the output to the screen will be when the code is run.

A. What is output by the following code?

```
int x1 = 0;
int y1 = ma(x1, 3) + x1;
System.out.print(x1 + " " + y1);

public static int ma(int x, int y) {
     x += 3;
     y--;
     return x * y;
}
```

B. What is output by the following code?

```
int x2 = -5;
int y2 = 10;
mb(x2, y2);
System.out.print(x2 + " " + y2);

public static void mb(int a, int b) {
     a = b / a;
     a--;
     b *= 2 + 3;
     System.out.print(a + " " + b + " ");
}
```

C. What is output by the following code?

```
int x3 = 3;
int y3 = 2;
double a3 = mc(x3, y3);
System.out.print(x3 + " " + y3 + " " + a3);

public static int mc(int x, int y) {
     y *= 2;
     x += 5;
     return x + y;
}

public static double mc(double x, double y) {
     x++;
     y--;
     return y * x + 1;
}
```

D. What is output by the following code?

```
int x4 = 2;
int y4 = 4;
int z4 = md(x4, md(y4, x4));
System.out.print(z4);

public static int md(int x, int y) {
     int z = y / x;
     x *= z;
     return x + y + z;
}
```

E. What is output by the following code?

```
String s51 = me(-2);
String s52 = me(3);
System.out.print(s51 + s52);

public static String me(int x) {
     String result = "e";
     for (int i = 0; i < x; i++) {
          result = result + result;
     }
     return result;
}
```

F. What is output by the following code?

```
int x6 = mf(2);
int y6 = mf(-2);
System.out.print(x6 + " " + y6);

public static int mf(int z) {
     int x = z * 3;
     z--;
     x /= 2;
     System.out.print(x + " " + z + " ");
     return z;
}
```

G. What is output by the following code?

```
int x7 = 4;
int y7 = 10;
System.out.print(mg(x7, y7) + " " + y7 + " " + mg(x7, x7));


public static int mg(int x, int z) {
     z += x;
     z = z - 5;
     x /= 2;
     System.out.print(x + " " + z + " ");
     z -= 3;
     return z;
}
```

4. Programming: 12 points - Complete the `printUTSquare` method. The method header is:

```
public static void printUTSquare(int size)
```

Examples of UT squares:

If size = 0, nothing is printed.

If size = 1 the following is printed:

```
U
```

If size = 2 the following is printed:

```
UU
UT
```

If size = 3 the following is printed:

```
UUU
UUT
UTT
```

If size = 3 the following is printed:

```
UUU
UUT
UTT
```

If size = 4 the following is printed:

```
UUUU
UUUT
UUTT
UTTT
```

If size = 5 the following is printed:

```
UUUUU
UUUUT
UUUTT
UUTTT
UTTTT
```

You may assume the parameter `size` is >= 0.

Complete the `printUTSquare` method on the next page.

```
public static void printUTSquare(int size) {
```

5. Programming: 12 points Write a method named `getProducts`. You must write the method header and the code for the method.

The method accepts one parameter, a `Scanner` object. Assume the `Scanner` object is connected to standard input.

The method prompts the users for the number of pairs the user will input. The method then asks the user for that number of pairs of numbers. For each pair the method prints out the product of those numbers.

After reading in the correct pairs of numbers, the method prints out the sum of the products and returns that sum to the code that called the method.

Example output of the method. The users input is bolded:

```
Enter the number of pairs: 3
Enter pair 1, 1st number: 5
Enter pair 1, 2nd number: 3
5 * 3 = 15
Enter pair 2, 1st number: 0
Enter pair 2, 2nd number: 10
0 * 10 = 0
Enter pair 3, 1st number: -2
Enter pair 3, 2nd number: 4
-2 * 4 = -8
Sum of products = 7
```

The method returns the sum of the products.

You may assume the use enters a integer > 0 for the number of pairs.

You may assume the user always enters an integer when asked.

Complete the entire `getProducts` method including the method header on the next page:

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

6. Programming: 12 points Write a static method, `diceGame`. The method header is

```
public static void diceGame(int numRolls)
```

The method simulates a dice game. In this game a **ten-sided die** whose faces are the numbers 0 through 9 is rolled a number of times equal to the parameter `numRolls`.

The player starts with 0 points. The die is rolled a number of times equal to the given parameter.

On a roll of 2 through 9, that many points are added to the player's point total.

On a roll of 1, the player's point total is doubled.

On a roll of 0, the player's point total becomes 0.

Consider the following output of the method. Assume `numRolls` equals 7

```
Roll 1 is 2. points = 2
Roll 2 is 7. points = 9
Roll 3 is 5. points = 14
Roll 4 is 0. points = 0
Roll 5 is 8. points = 8
Roll 6 is 1. points = 16
Roll 7 is 7. points = 23
```

You may assume `numRolls` is >= 1.

Recall the `Math.random()` method returns a double between 0 inclusive and 1.0 exclusive. Each possible double in that range has a roughly equal chance of being selected each time `Math.random` is called.

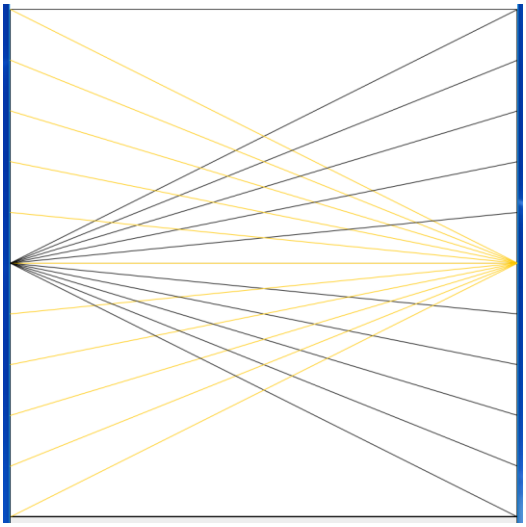
Complete the `diceGame` method on the next page.

```
public static void diceGame(int numRolls) {
```

7. Graphics Programming: 12 points Complete a method to produce the following starburst pattern. The parameters for the method are:

- The `Graphics` object for the `DrawingPanel`. You do not know what the current Color is.
- The `size` of the `DrawingPanel`. The `DrawingPanel` is square with width and height both equal to `size`. The drawing panel has already been created.
- The number of lines to draw on each side of the starburst. You may assume this value -1 is an integer divisor of the size of the `DrawingPanel`.

Here is the output of the method call `starburst(g, 800, 11);`



Note, (11 - 1) divides evenly into 800.

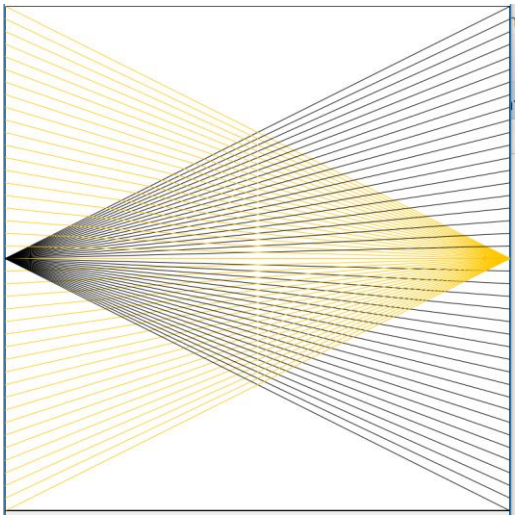
The vertical lines on the left and right are not part of the `DrawingPanel`.

The method draws lines equal to numLines from the middle of the left side spread out to the right side. These line are colored **black**.

The method draws lines equal to numLines from the middle of the right side spread out to the left side. These lines are colored **orange**.

The space between the lines on the spread side is always the same.

Here is the output of the method call `starburst(g, 800, 41);`



Note, (21 - 1) divides evenly into 800.

Note, nothing special is done for the interesting star like pattern in the middle to appear. That occurs simply by drawing the lines.

Recall the following methods from the Graphics class:

```
drawLine(int x1, int y1, int x2, int y2)

setColor(Color c)
```

Complete the method on the next page.

```
public static void starburst(Graphics g, int size, int numLines) {
```