

CS 312 – Exam 1 – Fall 2015

Name: SOLUTION SOLUTION SOLUTION

UTEID: SOLUTION SOLUTION SOLUTION _

Circle your TA's Name: Aila CK Jialin Katherine B
 KG Kris Megan Roman Sonika

Problem Number	Topic	Points Possible	Points Off
1	Expressions	18	
2	Code Tracing	16	
3	Method Trace	14	
4	Programming	10	
5	Programming	10	
6	Programming	15	
7	Graphics	15	
TOTAL POINTS OFF:			
SCORE OUT OF 98:			

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 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.
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, 18 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. $3 * 5 + 2 - 6 * 5$ -13
- B. $2 + 3 * 6 / (4 + 2) * 5$ 17
- C. $20 / 6 + 1 / 5 + 7 / 2$ 6
- D. $1.0 / 4.0 + 6 / 4 + 2.6 / 2$ 2.55
- E. $2.5 * 4 + 15.0 / 3.0$ 15.0
- F. $375 \% 100 / 7 * 2$ 20
- G. $4 * 6 \% 51 + 12 \% 4$ 24
- H. `"UT" + "horns" + 3` "UThorns3"
- I. $3 + 2 + "C!" + 1 + 15$ "5C!115"
- J. $2 + "RE" + 2 * 10$ "2RE20"
- K. `(double) (25 / 10 * 3)` 6.0
- L. `(int) (25 / 10.0 * 3 + 0.25)` 7
- M. `(int) (-7.0 / 2 * 3.1)` -10
- The Math methods `ceil`, `floor`, `pow`, and `abs` all return doubles.
- N. `Math.pow(2.0, 3.0)` 8.0
- O. `Math.max(20 / 10, 10 % 20) // returns an int` 10
- P. `Math.floor(3.99 * 4)` 15.0
- Q. `Math.ceil(-3.011)` -3.0
- R. `Math.abs(Math.ceil(-3.5) * Math.floor(6.8 / 2 * 3))` 30.0

2. Code tracing - 2 points each, 14 points total. Place your answer in the box to the right of the code. If the code results in a syntax or runtime error, state the kind of error that occurs.

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

```
int xa = 5;
int ya = xa;
xa = xa + 3 + ya;
System.out.print(xa + " " + ya);
```

13 5

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

```
int xb = 2;
int yb = 5;
int zb = xb * yb + 1;
yb -= zb;
xb++;
System.out.print(xb + " " + yb + " " + zb);
```

3 -6 11

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

```
int xc = 4;
int yc = 3;
yc *= 2 + xc;
xc -= xc * 2;
System.out.print(xc + " " + yc);
```

-4 18

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

```
double ad = 1.5;
int xd = 4;
int yd = 5;
double bd = yd / xd + ad - 3;
System.out.print(ad + " " + bd);
```

1.5 -0.5

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

```
int xe = -4;
int ye = 5;
for(int i = 0; i <= 4; i++) {
    xe += 2;
    ye = 5 * xe;
}
System.out.print(xe + " " + ye);
```

6 30

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 = 10; i <= 35; i++) {  
    System.out.print("*");  
}
```

26

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 ig = 0; ig < 8; ig++) {  
    for(int jg = 0; jg < 5; jg++) {  
        for(int kg = 0; kg < 10; kg++) {  
            System.out.print("*");  
        }  
    }  
}
```

400

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 ih = 2; ih < 6; ih++) {  
    for(int jh = 0; jh < ih; jh++) {  
        System.out.print("*");  
    }  
    System.out.print("*");  
    for(int jh = 10; jh >= 0 ; jh--) {  
        System.out.print("*");  
    }  
}
```

62

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.

Consider the following methods that are all part of the same program:

```
public static void a(int x, int y) {
    x = x * 3;
    x--;
    y = y - 3;
    System.out.print(x + " " + y + " ");
}
```

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

```
public static void c(int x) {
    if (x < 5) {
        x *= 3;
    } else {
        x = x / 3;
    }
    if(x % 2 == 1) {
        x += 2;
    }
    System.out.print(x);
}
```

```
public static int d(int i) {
    i++;
    i++;
    i = b(i + 1, i - 2);
    System.out.print(i);
    return i;
}
```

A. Given the methods above, what is output by the following code?

```
int xa = 3;
int ya = 2;
a(xa, ya);
```

8 -1

B. Given the methods above, what is output by the following code?

```
int xb = 4;
int yb = -2;
a(xb, yb);
System.out.print(xb + " " + yb);
```

11 -5 4 -2

C. Given the methods above, what is output by the following code?

```
int xc = -3;
int yc = 4;
yc = b(xc, yc);
System.out.print(xc + " " + yc);
```

-3 10

D. Given the methods on the previous page, what is output by the following code?

```
int xd = 45;
c(xd);
```

17

E. Given the methods on the previous page, what is output by the following code?

```
int xe = -3;
int ye = 7;
c(ye + xe + 1);
```

3

F. Given the methods on the previous page, what is output by the following code?

```
int xf = 4;
d(xf);
```

20

G. Given the methods on the previous page, what is output by the following code?

```
int xg = 3;
System.out.print("x is " + d(xg) + "!");
```

16x is 16!


```

public static void printFigure(int size) {
    final int NUM_LINES = size * size;
    for(int line = 1; line <= NUM_LINES; line++) {
        final int DOTS = NUM_LINES - line;
        for(int i = 0; i < DOTS; i++) {
            System.out.print(".");
        }
        for(int i = 0; i < line; i++) {
            System.out.print("*");
        }
        System.out.println();
    }
}

```

Criteria:

header, 1 point

number of lines correct, 1 point

outer loop correct: 2 points

prints correct number of dots: 3 points

prints correct number of stars: 2 points

println: 1 point

5. Programming - Write a static method named `printMultiples` that accepts an integer value and an integer `num` as parameters.

The method prints a complete line of output reporting the first `num` multiples of values.

Consider the following examples:

```
printMultiples(8, 2);  
prints: The first 2 multiples of 8 are 8, 16
```

```
printMultiples(4, 6);  
prints: The first 6 multiples of 4 are 4, 8, 12, 16, 20, 24
```

Assume the parameter `num` is ≥ 2 . value can be any integer value.

```
public static void printMultiples(int value, int num) {  
    System.out.print("The first " + num  
        + " multiples of " + value + " are ");  
  
    for(int i = 1; i < num; i++) {  
        System.out.print(value * i + ", ");  
    }  
    System.out.println(value * num); // or print  
}
```

header with parameters: 1 point

print first part, "The first ... ": 1 point

for loop: 3 points

print out multiples in correct order: 3 points

multiples of value calculated correctly: 1 point

no comma after last value handled correctly: 1 point

return statement in loop: -4

6. Programming - Write a method, `timesAllSame` that accepts a single parameter, an integer that specifies the number of times to flip a set of 3 coins. The method returns the number of times the 3 coins were flipped and were all the same, either all heads or all tails. Assume the integer parameter is greater than or equal to 0.

To simulate flipping a coin, use the `Math.random()` method:

```
public static double random()
```

Returns a double value with a, greater than or equal to 0.0 and less than 1.0. Returned values are chosen with (approximately) uniform distribution from that range.

Assume values < 0.5 are heads, and values >= 0.5 are tails.

Complete the `timesAllSame` below including the method header.

```
int total = 0;
for(int i = 0; i < numFlips; i++) {
    double c1 = Math.random();
    double c2 = Math.random();
    double c3 = Math.random();
    if(c1 < 0.5 && c2 < 0.5 && c3 < 0.5) {
        total++;
    } else if(c1 >= 0.5 && c2 >= 0.5 && c3 >= 0.5) {
        total++;
    }
}
return total;
}
```

header: 1 point

total variable: 1 point

loop correct: 3 points

call `Math.random` correctly 3 times: 3 points

correctly determine if all flips the same for a given set: 5 points

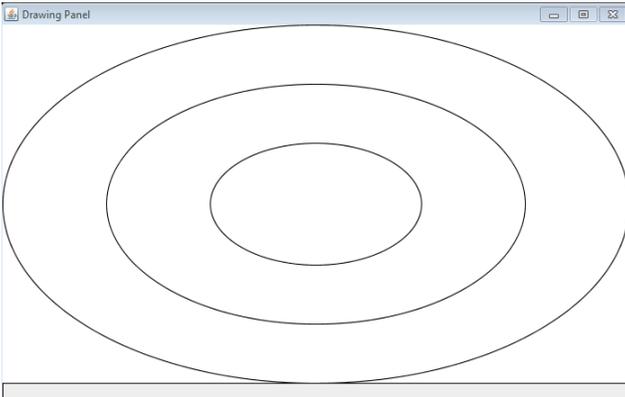
increment total correctly if appropriate: 1 point

return correct value: 1 point

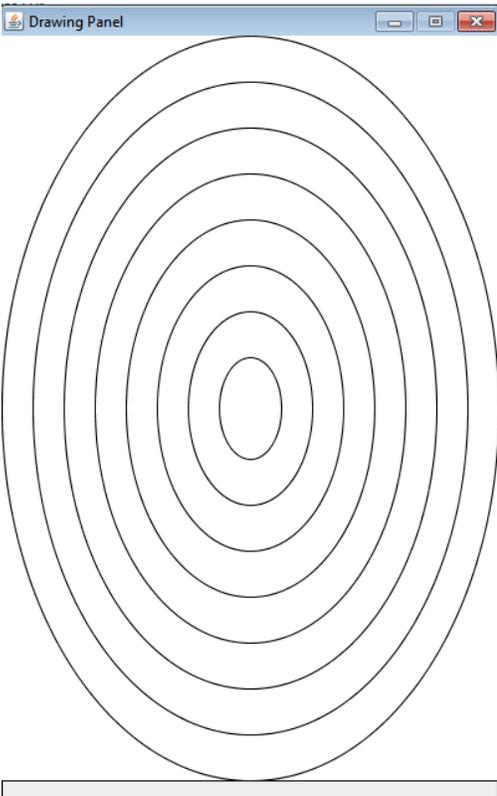
7. Graphics Programming. Complete a method to produce the following output of concentric circles. The parameters for the method are

- The `Graphics` object for the `DrawingPanel`. Color has already been set to `BLACK`
- The width and height of the drawing panel given as integers. The drawing panel has already been created.
- The number of circles to draw.

Here is the output of the method call `drawCircles(g, 700, 400, 3);`



Here is the output of the method call `drawCircles(g, 400, 600, 8);`



Complete the method on the next page.

```

public static void drawCircles(Graphics g, int width,
                               int height, int numOvals) {
    int x = 0;
    int y = 0;
    int ovalWidth = width;
    int ovalHeight = height;
    int widthFraction = width / (numOvals * 2);
    int heightFraction = height / (numOvals * 2);
    for(int i = 0; i < numOvals; i++) {
        g.drawOval(x, y, ovalWidth, ovalHeight);
        x += widthFraction;
        y += heightFraction;
        ovalWidth -= widthFraction * 2;
        ovalHeight -= heightFraction * 2;
    }
}

```

loop correct, correct number of ovals: 3 points

draw first oval correctly: 2 points

calculate x and y coordinates of each oval correctly: 4 points

calculate width and height of each oval correctly: 4 points

calls to draw oval correct, with correct parameters: 2 points

