

CS 312 – Midterm 1 – Fall 2014

Name: _____

UTEID: _____

Circle your TA's Name: Dory Tamara Eric
 Jose Stas Aaron

Problem Number	Topic	Points Possible	Points Off
1	Expressions	18	
2	Code Tracing	18	
3	Syntax Errors	10	
4	Writing Methods (Loop - Figure)	25	
5	Programming Using Graphics	25	
6	Programming	15	
7	Method Tracing	10	
TOTAL POINTS OFF:			
SCORE OUT OF 121:			

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 UTEID.

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. | <code>3 * 4 + 2 * 2 - 1</code> | _____ |
| B. | <code>2 / 4 + 6 / 4</code> | _____ |
| C. | <code>1.5 / .3 + 6 / 4</code> | _____ |
| D. | <code>25 % 8 + 1.5 * 2</code> | _____ |
| E. | <code>40 % 10 + 17 % 10</code> | _____ |
| F. | <code>"ut" + "cs" + 1 + "dh"</code> | _____ |
| G. | <code>2 + 3 + "cr" + 2 + 2</code> | _____ |
| H. | <code>"cp" + (3 * 3) + "x" + (2 / 4)</code> | _____ |
| I. | <code>4 - 6 + "gdc" + 5 + "1 + 2"</code> | _____ |
| J. | <code>3 * 2 * 4 / 5 % 2</code> | _____ |
| K. | <code>0 % 10 + 5 / 2 + 1.5</code> | _____ |
| L. | <code>1 + 5 * 0 + "hi" + 1.5</code> | _____ |
| M. | <code>1.5 + 2 * 3</code> | _____ |

The Math methods `ceil`, `floor`, `sqrt`, `pow`, and `abs` all return doubles.

- | | | |
|----|--|-------|
| N. | <code>Math.abs(-3.5) + Math.max(2.5 + 2, 3.5 * 2)</code> | _____ |
| O. | <code>Math.ceil(-1.75) + Math.floor(2.5)</code> | _____ |
| P. | <code>Math.min(Math.max(.5, 1), Math.min(1.5, 0.5))</code> | _____ |
| Q. | <code>Math.pow(2, 4)</code> | _____ |
| R. | <code>(int) (1.99 * 5) + (double) 6 / 4</code> | _____ |

2. Code tracing. 2 points each, 18 points total. Place your answer in the box to the right of the code. If the code results in a compiler or runtime error, state the kind of error that occurs.

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

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

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

```
int xb = 3;
xb++;
xb = xb + 2;
double ab = xb / 4;
ab -= 2 * xb;
System.out.print(xb + " " + ab);
```

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

```
int xc = 3;
for(int i = 1; i <= 3; i++) {
    int yc = i * 2;
    xc = xc + yc;
}
System.out.print(xc);
```

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

```
double ad = 1.5;
double bd = -ad;
ad *= 3;
bd = -1 - bd;
System.out.print(ad + " " + bd);
```

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

```
String se = "xe";
int xe = 3;
int ye = 4;
xe++;
ye = xe;
xe = ye;
se = ye + se + xe;
System.out.print(xe + " " + se);
```

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 = -5; i <= 3; i++) {  
    System.out.print("*");  
}
```

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 = 1; ig <= 8; ig++) {  
    for(int jg = 5; jg > 0; jg--) {  
        System.out.print("*");  
        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 ih = 0; ih < 4; ih++) {  
    System.out.print("*");  
    for(int jh = 0; jh < ih; jh++) {  
        System.out.print("*");  
    }  
    System.out.print("*");  
    for(int jh = 0; jh < 3; jh++) {  
        System.out.print("*");  
    }  
    System.out.print("*");  
}
```

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

```
double ai = 1.7;  
double bi = -2.5;  
if(ai <= Math.abs(bi))  
    System.out.print("A");  
if(Math.pow(1.7, 5.0) > 1)  
    System.out.print("B");  
if(Math.floor(ai) > Math.ceil(bi))  
    System.out.print("C");  
if(Math.floor(bi) < Math.ceil(bi))  
    System.out.print("D");
```

3. Syntax errors. 10 points. Each of the following code snippets contains a syntax error. Explain what the syntax error is in a single sentence.

A.

```
for(int final = 0; final < 10; final++) {  
    int x = 10;  
    System.out.print(x++ * 10);  
}  
// What causes the syntax error?
```

B.

```
String st = "";  
int x = 5;  
int y = -3;  
st = x * y + "res-" + x - y + "x" + "\\n";  
System.out.print(st + x);  
// What causes the syntax error?
```

C.

```
int yg;  
yg = 3;  
int 2x = yg * 10;  
System.out.print(Math.pow(yg, 2x));  
// What causes the syntax error?
```

D.

```
int m = 3;  
int n;  
int _o = 5;  
m *= n + _o;  
System.out.print(m);  
// What causes the syntax error?
```

E.

```
double ah = 12;  
int xh = ah / 4;  
double bh = xh * 3;  
for(int i = 0; i < ah; i++)  
    System.out.print("***");  
// What causes the syntax error?
```


4. Programing and Loops. 20 points. Write a Java method to produce the following output. The method relies on a parameter named `size`.

When the parameter `size` is 3 the output is:

```
--*--  
-***-  
*****
```

When the parameter `size` is 5 the output is:

```
-----*-----  
----***----  
--*****--  
-*****-  
*****
```

Complete your method, including the method header, in the space provided:

More room for question 4 on next page.

More room for question 4 if necessary.

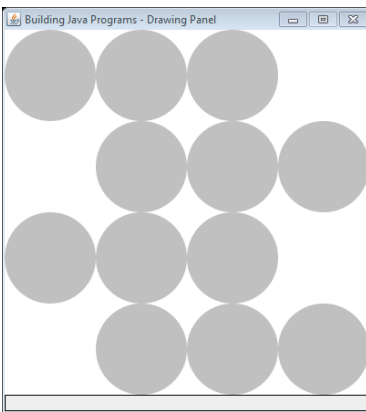
5. Graphics Programming. 20 Points. Complete a method to produce the following output. The parameters for the method are

- The `Graphics` object for the `DrawingPanel`.
- The size of the `DrawingPanel`. Assume the `DrawingPanel` is square with width and height equal to the size parameter.
- The number of rows. Assume that the size of the `DrawingPanel` is a multiple of the number of rows.
- The number of circles to skip per row.

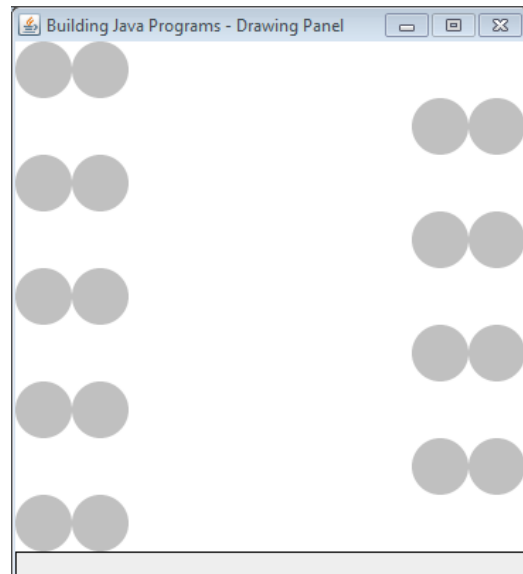
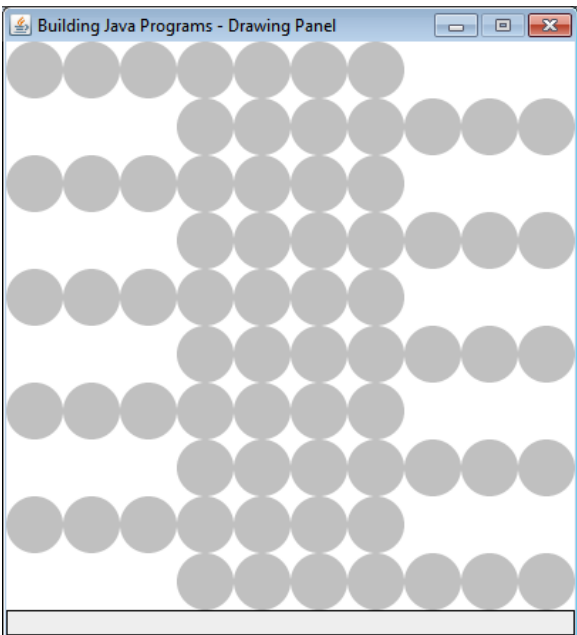
Your method must be general and work for various values of the size of the `DrawingPanel` and different values for the number of rows and the number of circles to skip per row.

The circles are filled with the `LIGHT_GRAY` color from the `Color` class.

Here is the output of the method call `drawFigure(g, 400, 4, 1);`



The output of the method call `drawFigure(g, 400, 10, 3)` is shown below on the left and the output of the method call `drawFigure(g, 360, 9, 7)` is shown on the right.



Complete the method on the next page.

```
public static void drawFigure(Graphics g, int size, int numRows, int skip)
{
```

6. Programming. 15 points. Write a method named `soccerWinner` that determines which soccer team wins a *two-legged tie*. A two-legged tie is commonly used in soccer tournaments, although not the World Cup.

The two-legged tie consist of two games between teams A and B.

The first game (or leg) is played at team A's home stadium and the second game (or leg) is played at team B's home stadium.

The winner of the two-legged tie is based on aggregate goals, the total number of goals scored in the two games.

Consider the following example:

- First leg at Team A's home stadium: Team A 4 – 1 Team B
- Second leg at Team B's home stadium: Team B 2–1 Team A

The aggregate score is Team A 5 - 3 Team B, so Team A wins the two-legged tie.

If there is a tie in aggregate score the first tiebreak is based on *away goals*. In other words if the aggregate score is tied the team that scored more goals away from their home stadium is the winner.

Consider the following example

- First leg at Team A's home stadium: Team A 4 – 2 Team B (Team B scored 2 away goals)
- Second leg at Team B's home stadium: Team B 2 – 0 Team A (Team A scored 0 away goals)

The aggregate score is tied at Team A 4 - 4 Team B, but the away goals are Team A 0 - 2 Team B so Team B wins the two-legged tie.

If there is still a tie after considering away goals then the teams complete a shootout with penalty kicks and whichever team scores more penalty kicks wins.

Your method shall accept 4 parameters: Team A's number of goals at home, Team A's number of goals away, Team B's number of goals at home, and Team B's number of goals away.

Based on these parameters if there is a winner the method prints out the winner.

If the there is still a tie the method asks the user for the number of penalty kicks Team A and Team B scored and prints out the winner based on that information.

Examples of **output** for the `soccerWinner` method are shown below

parameters are Team A home goals, Team A away goals, Team B home goals, Team B away goals

```
soccerWinner(4, 1, 2, 1) -> Team A wins
soccerWinner(4, 0, 2, 2) -> Team B wins
```

The user input is italicized below:

```
soccerWinner(2, 1, 2, 1) ->
Shootout
Enter Team A penalty kicks: 3
Enter Team B penalty kicks: 4
Team B wins
```

Assume the `Scanner` class has already been imported to the program.

You may assume the user will enter valid input for the penalty kicks if necessary. In other words the user will enter integer values greater than or equal to 0 and the two values will not be equal to each other.

Complete your `soccerWinner` method, including the method header, below.

7. Method Tracing and Parameters Simulation. 10 points.

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

```
public static void a(int x, int y) {
    int z = x;
    x *= 2;
    y /= 2;
    System.out.print(x + " " + y + " " + z);
    x = z - 2;
}

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

public static int c(int x, int y) {
    int r = b(x, y);
    int s = b(y, x);
    r += 2 + x;
    s = y - s;
    return r + s;
}

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


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

```
int xa = 2;
int ya = 3;
a(xa, ya);
System.out.print(xa + " " + ya);
```



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

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



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

```
int xc = 2;  
int yc = 1;  
int zc = b(xc, yc) + c(yc, xc);  
System.out.print(xc + " " + yc + " " + zc);
```

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

```
int xd = 3;  
int yd = 2;  
int zd = d(xd + yd);  
xd = b(zd, yd);  
System.out.print(xd + " " + yd + " " + zd);
```

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

```
int ze = 3;  
int xe = d(ze);  
System.out.print(ze + " " + d(xe) + " " + b(xe, ze) + " " + d(ze));
```

