

CS 312 – Midterm 1 – Fall 2013

Your Name: SOLUTION SOLUTION SOLUTION

Your UTEID: SOLUTION SOLUTION SOLUTION

Circle your TA's Name: EDAENA LEIF MUBASHIR

Problem Number	Topic	Points Possible	Points Off
1	Expressions	18	
2	Code Tracing	18	
3	Tracing Graphics Programs	10	
4	Writing Methods (Loop - Figure)	20	
5	Programming Using Graphics	20	
6	Programming - Conditionals	15	
7	Method Tracing	10	
8	Programming - Return Methods	10	
TOTAL POINTS OFF:			
SCORE OUT OF 121:			

Instructions:

1. Please turn off your cell phones
2. There are 8 questions on this test.
3. You have 2 hours to complete the test.
4. You may not use a calculator.
5. Please make your answers legible.
6. When code is required, write Java code.
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. The exam is worth 121 points. Grades will be scaled to 150 for gradebook.

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.

ANSWER AS SHOWN OR -1

A.	$4 * 5 + 100 / 20 + 3 * 2$	31
B.	$17 / 8 + 3$	5
C.	$60 / ((3 + 2) * 3)$	4
D.	$60 / 3 + 2 * 3$	26
E.	$1.5 + 3 / 2$	2.5
F.	$4.5 / 1.5$	3.0
G.	$27 \% 5$	2
H.	$7 \% 33 * 3 + 8 \% 24$	29
I.	$1050 \% 10$	0
J.	$127 \% 10 + 3 * 2.0$	13.0
K.	$5 * 6 + (12 \% 2 * 7)$	30
L.	<code>"CS" + "CNS" + "12"</code>	<code>"CSCNS12"</code> (-1 if no quotes)
M.	<code>"JAVA" + "java"</code>	<code>"JAVAJava"</code> (-1 if no quotes)
N.	<code>"UT" + (3.7 + 1.2)</code>	<code>"UT4.9"</code>
O.	$5 - 3 + \text{"GDC"} + 6 + 3$	<code>"2GDC63"</code>

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

P.	<code>Math.sqrt(25) + 3</code>	8.0
Q.	<code>Math.floor(-1.2) + Math.ceil(3.85)</code>	2.0
R.	<code>Math.abs(0) + Math.abs(-4.5)</code>	4.5

2. Code tracing. 2 points each, 18 points total. Place your answer in the box to the right of the code
GRADING: -1 FIRST MISTAKE, -2 SECOND MISTAKE PER PART

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

```
int xa = 2;  
xa = 10;  
xa = xa - 5;  
xa++;  
System.out.print(xa);
```

6

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

```
int xb = 5;  
int yb = 3;  
int zb = xb + yb;  
xb = zb + xb;  
zb = 3 * yb;  
System.out.print(xb + " " + yb + " " + zb);
```

13 3 9

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

```
int xc = 7;  
int yc = 5;  
yc = xc;  
xc = yc;  
System.out.print(xc + " " + yc);
```

7 7

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

```
int xd = -2;  
int yd = 3;  
yd += xd * yd + 2;  
System.out.print(xd + " " + yd);
```

-2 -1

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

```
int xe = 3;  
int ye = 10;  
int ze = 2;  
ze *= ye - xe * 2;  
System.out.print(ze);
```

8

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

50

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 = -2; ig <= 6; ig++) {  
    for(int jg = 1; jg <= ig; jg++) {  
        System.out.print("*");  
    }  
}
```

21

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 < 3; ih++) {  
    System.out.print("*");  
    for(int jh = 1; jh <= 5; jh++) {  
        System.out.print("*");  
    }  
    System.out.print("*");  
    for(int jh = 0; jh < 10; jh++) {  
        System.out.print("*");  
    }  
}
```

85

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

```
int xi = 3;  
int yi = 5;  
if(xi > yi)  
    System.out.print("A");  
if(xi * xi > yi)  
    System.out.print("B");  
else  
    System.out.print("C");  
if(xi % yi == 0 || yi % xi == 0)  
    System.out.print("D");  
else  
    System.out.print("E");
```

BE

3. Tracing Graphics Programs. 10 points. Sketch the DrawingPanel window that is produced when the following program when is run. Do not draw the title bar.

```
import java.awt.Color;
import java.awt.Graphics;

public class Draw {

    public static final int SIZE = 400;

    public static void main() {
        DrawingPanel p = new DrawingPanel(SIZE, SIZE);
        Graphics g = p.getGraphics();
        int part = SIZE / 4;

        // parameters for drawLine are x1, y1, x2, y2, end points of line
        int value = part;
        for(int i = 0; i < 4; i++) {
            g.drawLine(0, 0, SIZE, value);
            g.drawLine(0, 0, value, SIZE);
            value += part;
        }

        // parameters for fillRect are x, y, width, height
        g.setColor(Color.ORANGE);
        g.fillRect(0, 0, part, part);
    }
}
```

ORANGE ---->

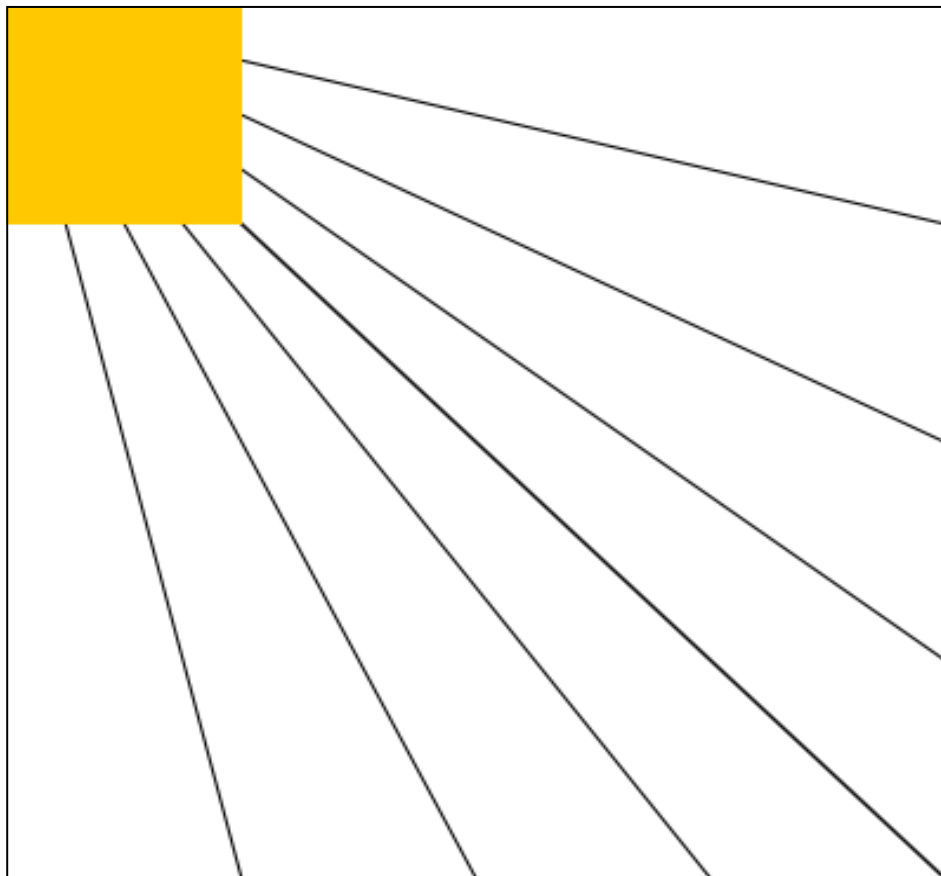
7 lines: 3 points
(okay if main
diagonal think ,
two close):

spreads from top
left out: 3 points

square in top left
approximately
right size: 2 points

square shaded or
indiatced full: 1

color on square
labeled orange: 1
point



4. Programing and Loops. 20 points. Write a Java method to produce the following output. The output 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:

Suggested Solution:

```
public static void starsAndStripes(int size) {
    for(int line = 1; line <= size; line++) {
        int numStars = line * 2;
        for(int i = 0; i < numStars; i++) {
            System.out.print("*");
        }
        int numPipes = size + 2 - line;
        for(int i = 0; i < numPipes; i++) {
            System.out.print("|");
        }
        System.out.println();
    }
}
```

General Grading Criteria:

method header: 1

size parameter: 2

outer loop for lines: 2 points

calculate number of stars correct (can be in loop test): 3 points

inner loop for stars: 2 points

calculate number of pipes correct (can be in loop test): 4 points

inner loop for pipes: 2 points

print not println for stars and pipes: 2 points

println to end line: 2 points

Other deductions:

1. main method: -1
2. infinite loop -8
3. hard coded numbers for size, 6 to 8 depending on severity

Common Errors:

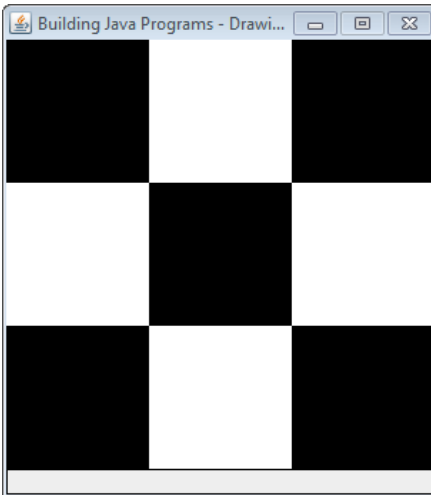
1. no method or parameter
2. not okay to make main the method, parameters wrong
3. number of pipes always the same in each line. The number of pipes goes down by one in each line
4. Python like code for "multiplying" a String. This does not work in Java.

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`, and the size of the small squares. Assume the `DrawingPanel` is square with length and width equal to the `size` parameter. Assume that the size of the `DrawingPanel` is a multiple of the small square size. In other words $(\text{DrawingPanel size}) \% (\text{square size}) == 0$.

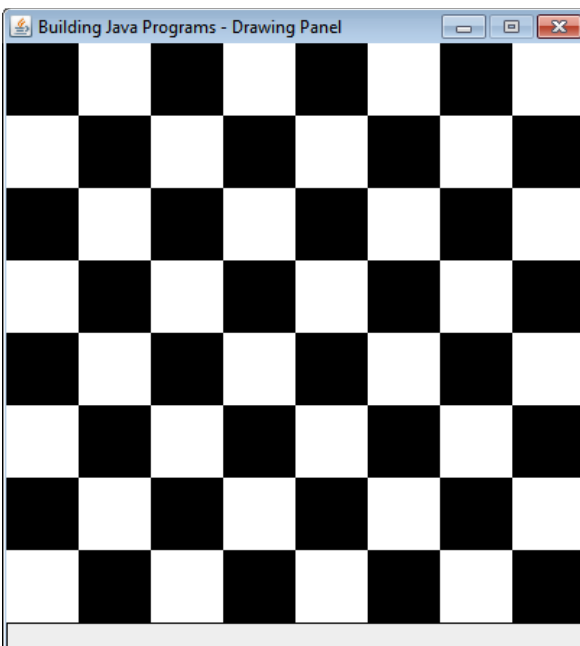
Your method must be general and work for various values of the size of the `DrawingPanel` and various sizes of the small squares, assuming the size of the `DrawingPanel` is a multiple of the size of the small square.

Assume the color of the `Graphics` object has already been set to `Color.BLACK`.

Here is the output of the method call `drawBoard(g, 300, 100);`



Here is the output of the method call `drawBoard(g, 400, 50);`



Complete the method on the next page.

Complete the following method:

```
// Assume:
// 1. size is the drawing panel width and height
// 2. size % squareSize == 0

public static void drawBoard(Graphics g, int size, int squareSize) {
    // not necessary for answers, but in case background not white:
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, size, size);
    g.setColor(Color.BLACK);

    // EXPECTED STUDENT ANSWER START HERE
    int numRows = size / squareSize;
    int y = 0;
    for(int row = 0; row < numRows; row++) {
        int x = 0;
        if(row % 2 == 1) {
            x = squareSize; // odd rows start x over a square
        }
        int numBoxes = numRows - row % 2;
        for(int box = 0; box < numBoxes; box++) {
            g.fillRect(x, y, squareSize, squareSize);
            x += 2 * squareSize;
        }
        y += squareSize;
    }
}
```

General Grading Criteria:

Calculate number of rows (and thus columns), can be in context of loop test: 3 points

Outer loop for rows (or columns): 2 points:

Inner loop for boxes in row (or column): 3 points

calculate starting x coordinate for row (or y for column) correctly: 4 points

calculate and draw the correct number of black boxes per row: 2 points

correctly fill rectangles of appropriate size: 2 points (-1 for drawRect)

alter x correctly for next rectangle in row (or y for columns): 2 points

alter y after inner loop if doing rows or x if doing columns: 2 points

Hardest question on the test. Lots of different solutions.

- okay to draw white squares
- okay to draw off visible array
- don't create new DrawingPanel
- lots of people didn't correctly alternate white / black.
- not okay to just toggle black / white. row ends in white, next cell (first in next row) also white.
- lots of people just drew a diagonal of squares, not the rows. Didn't track y.

6. Programming. 15 points. Write a method named `speedingTicket` that determines if a driver will get a speeding ticket from a police officer and if so, the amount of the fine for the ticket.

The method accepts 3 parameters: the speed limit and the speed of the car, in miles per hour. Both of these values are integers. The third parameter is the number of cups of coffee the police officer has consumed this morning, a floating point number.

The police officer will give a ticket to the speeding driver if any of the following conditions are met:

- the police officer has had less than 2.5 cups of coffee and the driver's speed is greater than 5 miles per hour over the speed limit
- the police officer has had 2.5 cups of coffee or more and the driver's speed is greater than 10 miles per hour over the speed limit
- the driver's speed is 100 miles per hour or more regardless the speed limit or the of number of cups of coffee the police officer has had

If the police officer does not give the driver a ticket then they give the driver a warning.

The method prints out if the police officer gives the driver a ticket or a warning. If the driver gets a ticket the method prints out the amount of the fine. The amount of the fine is \$50 plus \$10 for every mile above the speed limit. If the driver is going twice the speed limit or more the final fine is doubled.

Examples of output from various calls to the `speedingTicket` method. Parameters are the speed limit, the driver's speed, and number of cups of coffee the police officer has had this morning.

```
// parameters are speed limit, actual speed, and cups of coffee
speedingTicket(55, 60, 0.0) output warning
speedingTicket(55, 61, 0.0) output ticket! fine = $110
speedingTicket(55, 60, 2.5) output warning
speedingTicket(55, 61, 2.5) output warning
speedingTicket(55, 61, 5.0) output warning
speedingTicket(25, 49, 0.0) output ticket! fine = $290
speedingTicket(25, 50, 0.0) output ticket! fine = $600
speedingTicket(25, 51, 3.5) output ticket! fine = $620
speedingTicket(25, 51, 0.0) output ticket! fine = $620
speedingTicket(98, 100, 3.5) output ticket! fine = $70
speedingTicket(98, 100, 0.0) output ticket! fine = $70
speedingTicket(75, 100, 3.5) output ticket! fine = $300
```

Complete the method, including the method header, on the next page.

Complete your `speedingTicket` method, including the method header, below:

```
public static void speedingTicket(int limit, int speed, double cups) {
    int overLimit = speed - limit;
    if(speed >= 100 || (cups < 2.5 && overLimit > 5)
        || (cups >= 2.5 && overLimit > 10)) {

        int fine = 50 + 10 * overLimit
        if(speed >= limit * 2) {
            fine *= 2;
        }
        System.out.println("ticket! fine = $" + fine);
    }
    else {
        System.out.println("warning");
    }
}
```

header: 1 point

parameters: 1 points

basic if else structure for ticket or warning: 2 points

cases for ticket:

over 100: 1 point

diff > 5 and not enough coffee: 2 points

diff > 10 and enough coffee: 2 points

calculate basic fine correctly: 2 points

fine doubled case: 2 point

print correct message: 2 points

Common problems:

1. Not using if/ else if / else if / else if / else structure which results in multiple output in some cases
2. Not doubling the fine the actual speed is twice the speed limit.
3. if else statements - very confusing.

7. Method Tracing and Parameters Simulation. (2 points each)

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

```
public static void a(int x, int y) {  
    x++;  
    y = y - x + 1;  
    System.out.println(x + " " + y);  
}
```

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

```
public static void c(int x, int y) {  
    x += 2;  
    y -= 2;  
    System.out.println(x + " " + y);  
    a(x, y);  
}
```

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

GRADING - -1 for one mistake, -2 for 2 or more

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

```
int xa = 3;  
int ya = 8;  
a(xa, ya);  
System.out.println(xa + " " + ya);
```

4 5
3 8

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

```
int xb = 5;  
int yb = -4;  
xb = b(yb, xb);  
System.out.println(xb + " " + yb);
```

-6 -4

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

```
int xc = 2;  
int yc = 3;  
b(xc, yc);  
System.out.println(xc + " " + yc);
```

2 3

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

```
int xd = -2;
int yd = 2;
c(xd, yd);
System.out.println(xd + " " + yd);
```

```
0 0
1 0
-2 2
```

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

```
int xe = 5;
xe = d(xe) + 2;
System.out.println(xe + d(xe));
```

```
10 24 36
```

8. Programming. 10 points. Write a method that determines the sums of the volumes of a cube and a sphere based on two parameters: the side length of the cube and the diameter of the sphere.

Recall the volume of a cube = $(length\ of\ a\ side)^3$ and the volume of a sphere = $\frac{4}{3}\pi r^3$

The method takes two parameters, the length of a side of the cube and the diameter of the sphere.

For example given a side length of 3 and a diameter of 6 the method would return 140.09733552923257.

Given a side length of 1 and a diameter of 2 the method would return 5.1887902047863905.

Write the entire method, including the method header with parameters in the space provided. You may assume the side length of the cube and diameter of the sphere are both greater than or equal to 0. Use the `Math.PI` constant for the value of π .

```
public static double sumOfVolumes(double side, double diameter)
{
    double radius = diameter / 2;
    double cubeVolume = side * side * side;
    double sphereVolume = 4.0 / 3 * Math.PI
                          * radius * radius * radius;
    return cubeVolume + sphereVolume;
}
```

Method header: 1 point

parameters: 1 point (int or double)

cube volume: 2 points (Math.pow okay IF correct order (base, exponent))

sphere volume basic: 2 points

4 / 3 not int division: 1 point

Math.PI: 1 point

use radius not diameter: 1 point

return sum correctly: 1 point