

CS 312 – Midterm 1 – Fall 2014

Name: SOLUTION SOLUTION SOLUTION

UTEID: SOLUTION SOLUTION SOLUTION _

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. $3 * 4 + 2 * 2 - 1$ 15
- B. $2 / 4 + 6 / 4$ 1
- C. $1.5 / .3 + 6 / 4$ 6.0
- D. $25 \% 8 + 1.5 * 2$ 4.0
- E. $40 \% 10 + 17 \% 10$ 7
- F. `"ut" + "cs" + 1 + "dh"` `"utcs1dh"`
- G. $2 + 3 + "cr" + 2 + 2$ `"5cr22"`
- H. `"cp" + (3 * 3) + "x" + (2 / 4)` `"cp9x0"`
- I. $4 - 6 + "gdc" + 5 + "1 + 2"$ `"-2gdc51 + 2"`
- J. $3 * 2 * 4 / 5 \% 2$ 0
- K. $0 \% 10 + 5 / 2 + 1.5$ 3.5
- L. $1 + 5 * 0 + "hi" + 1.5$ `"1hi1.5"`
- M. $1.5 + 2 * 3$ 7.5

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

- N. `Math.abs(-3.5) + Math.max(2.5 + 2, 3.5 * 2)`
10.5
- O. `Math.ceil(-1.75) + Math.floor(2.5)` 1.0
- P. `Math.min(Math.max(.5, 1), Math.min(1.5, 0.5))`
0.5
- Q. `Math.pow(2, 4)` 16.0
- R. `(int) (1.99 * 5) + (double) 6 / 4` 10.5

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);
```

9 -3

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);
```

6 -11.0

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);
```

15

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);
```

4.5 0.5

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);
```

4 4xe4

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("*");  
}
```

9

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("*");  
    }  
}
```

80

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("*");  
}
```

30

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");
```

ABCD

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?
Cannot use final as an identifier, it is a reserved word.
```

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?
The subtract operator is not overridden for Strings.
```

C.

```
int yg;
yg = 3;
int 2x = yg * 10;
System.out.print(Math.pow(yg, 2x));
// What causes the syntax error?
2x is not a valid identifier.
```

D.

```
int m = 3;
int n;
int _o = 5;
m *= n + _o;
System.out.print(m);
// What causes the syntax error?
n has not been initialized.
```

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?
Cannot store an expression that results in a double in an int var. (Loss of precision.)
```

4. Programming 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:

```
public static void q4Figure(int size) {  
    for(int line = 0; line < size; line++) {  
        int dashes = size - 1 - line;  
        for(int j = 0; j < dashes; j++) {  
            System.out.print("-");  
        }  
        int numStars = line * 2 + 1;  
        for(int j = 0; j < numStars; j++) {  
            System.out.print("*");  
        }  
        for(int j = 0; j < dashes; j++) {  
            System.out.print("-");  
        }  
        System.out.println();  
    }  
}
```

header: 2 points

outer loop correct: 6 points

first set of dashes: 4 points

stars correct: 7 points

second set of dashes: 4 points

println: 2 points

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

```
public static void drawFigure(Graphics g, int size,  
                               int numRows, int circlesToSkip) {  
  
    g.setColor(Color.LIGHT_GRAY);  
    int circleSize = size / numRows;  
    int circlesPerRow = numRows - circlesToSkip;  
    int y = 0;  
    for(int row = 0; row < numRows; row++) {  
        int x = circleSize * circlesToSkip * (row % 2);  
        // okay to use if statement of course  
        for(int i = 0; i < circlesPerRow; i++) {  
            g.fillOval(x, y, circleSize, circleSize);  
            x += circleSize;  
        }  
        y += circleSize;  
    }  
}
```

set color: 2 points

circle size calculation: 4 points

outer loop for rows or columns: 3 points

set x correctly for skip or no skip: 4 points

inner loop for columns or rows: 3 points

fill oval correctly with correct size: 3 points

change x correctly: 3 points

change y correctly: 3 points

new DrawingPanel = -2

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.

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.

```
public static void soccerWinner(int teamAHome,
    int teamAAway, int teamBHome, int teamBAway) {
    int aTotal = teamAHome + teamAAway;
    int bTotal = teamBHome + teamBAway;
    if(aTotal > bTotal)
        System.out.println("Team A wins");
    else if(bTotal > aTotal)
        System.out.println("Team B wins");
    else if(teamAAway > teamBAway)
        System.out.println("Team A wins");
    else if(teamBAway > teamAAway)
        System.out.println("Team B wins");
    else {
        Scanner key = new Scanner(System.in);
        System.out.print("Enter Team A penalty kicks: ");
        int apk = key.nextInt();
        System.out.print("Enter Team B penalty kicks: ");
        int bpk = key.nextInt();
        if(apk > bpk)
            System.out.println("Team A wins");
        else
            System.out.println("Team B wins");
    }
}
```

header and parameters correct: 1 point

aggregate wins cases handled correctly: 5 points

away goals cases handled correctly: 4 points

penalty kicks handled correctly including creating `Scanner` and prompting for input: 5 points

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);
```

4 1 22 3

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);
```

-12 6

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);
```

2 1 3

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);
```

8 28 2 6

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));
```

4 2 4 3 0 -3 2