

CS312 Fall 2016 Exam 1 Solution and Grading Criteria.

Grading acronyms:

AIOBE - Array Index out of Bounds Exception may occur

BOD - Benefit of the Doubt. Not certain code works, but, can't prove otherwise

Gacky or Gack - Code very hard to understand even though it works. (Solution is not elegant.)

LE - Logic error in code.

NAP - No answer provided. No answer given on test

NN - Not necessary. Code is unneeded. Generally no points off

NPE - Null Pointer Exception may occur

OBOE - Off by one error. Calculation is off by one.

RTQ - Read the question. Violated restrictions or made incorrect assumption.

1. Expressions:

A.	$2 + 20 / 5 - 3 * 4 / 2$	0
B.	$15 / 10 + 20 / 30 + 3 / 2$	2
C.	$15 \% 5 + 23 \% 10 + 18 \% 12$	9
D.	$5.0 / 2.0 + 10.0 / 4.0$	5.0
E.	$4 + 3 / (1.0 * 2) / 0.5$	7.0
F.	$9 / 2.0 + 7 / 3 - 3.0 / 2$	5.0
G.	$9 + 5 * -3 + 4 + "X" + 4 + 2$	"-2X42"
H.	$"3" + 21 \% 20 + 2 \% 42$	"312"
I.	$9 \% 5 * 6 + 18 / 6 - 1$	26
J.	$15 / 20 + 20 / 15 + 8 / 10$	1
K.	$3 * 5 + 2 + "!!" + 2 * 3 + 1$	"17!!61"
L.	$(int) (.75 * 10)$	7
M.	$((int) .999) * 10 + 2.5$	2.5
N.	$"2 * 3 " + 2 * 3 + 2$	"2 * 3 62"
O.	$177 \% 100 \% 10 / 2$	3
P.	$Math.pow(2.0, 4.0)$	16.0
Q.	$Math.abs(-3.33 * -10.0)$	33.3
R.	$Math.floor(-1.3 * 3)$	-4.0

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**.

AS SHOWN or - 2. First two instances of "answer" counter wrong.

- A. 2 16
- B. wag3dog6 6
- C. 3.0 12
- D. 1.5 10
- E. Runtime error occurs
- F. 40
- G. 16
- H. 315
- I. 560

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.

-2 first two sets of quotes

- A. 0 -10-3 -9 (space between -10 and -3 okay)
- B. 7 43 2 (space between 4 and 3 okay)
- C. 17.05.5 5 (space between 17.0 and 5.5 okay)
- D. 3 50 45
- E. 344 11 (3 4 4 11 okay)
- F. 15
- G. 654535 10 25 (no extra spaces allowed)

4. Programming: 5 points -

```
public static double simpleGPA(int as, int bs, int cs) {  
    double num = as * 4.0 + bs * 3.0 + cs * 2.0;  
    int denom = as + bs + cs;  
    return num / denom;  
}
```

Method header correct: 1 point (parameters must be int)

calculate numerator correctly: 1 point

calculate denominator correctly: 1 point

perform division: 1 point (int division -1)

return: 1 point

any output -1

5. Programming: 12 points

```
public static void printPowers(int base, int num) {  
    int total = base;  
    System.out.print("The first " + num + " powers of ");  
    System.out.print(base + " are " + base + ", ");  
    final int LIMIT = num - 2;  
    for (int i = 0; i < LIMIT; i++) {  
        total *= base;  
        System.out.print(total + ", ");  
    }  
    System.out.println(total * base); // last one  
}
```

Method header correct: 1 point

Print first part (The first num powers of base are: 2 point

loop to print powers: 2 points

correctly calculate powers (could be nested loop): 4 points (lose this if use Math.pow)

correctly print powers: 1 points

handle case with no comma after last power: 2 points

6. Programming: 13 points Complete the method `closerPoint`. The method header is:

```
public static void closerPoint(int x1, int y1, int x2, int y2,
    int x3, int y3) {
    double d2 = distance(x1, y1, x2, y2);
    double d3 = distance(x1, y1, x3, y3);
    if (d2 < d3) {
        System.out.println("Point 2 is closer to Point 1");
    } else if (d3 < d2) {
        System.out.println("Point 1 is closer to Point 2");
    } else {
        System.out.print("Point 2 and Point 3 are equidistant from Point 1");
    }
}

public static double distance(int x1, int y1, int x2, int y2) {
    double part1 = (x1 - x2) * (x1 - x2); // or = Math.pow(x1 - x2, 2);
    double part2 = (y1 - y2) * (y1 - y2);
    return Math.sqrt(part1 + part2);
}
```

Note necessary to have separate method for distance.

Distance calculation correct: 5 points (partial credit possible)

Handle case when point 2 closer correctly: 3 points (partial credit possible)

Handle case when point 3 closer correctly: 3 points (partial credit possible)

Handle case when points equidistance correctly: 2 points (partial credit possible)

7. Graphics Programming: 20 points Complete a method to produce the following output.

```
public static void drawStaircase(Graphics g, int size, int numSteps) {
    int squareSize = size / numSteps;

    // draw the squares
    int x = 0;
    int y = 0;
    for (int i = 0; i < numSteps; i++) {
        g.fillRect(x, y, squareSize, squareSize);
        x += squareSize;
        y += squareSize;
    }

    // draw the circles
    y = squareSize;
    for (int row = 1; row < numSteps; row++) {
        x = 0;
        for (int i = 0; i < row; i++) {
            g.fillOval(x, y, squareSize, squareSize);
            x += squareSize;
        }
        y += squareSize;
    }
}
```

Calculate square / circle size correctly: 3 points

draw squares

- draw some squares: 1 point
- draw correct number of squares: 1 point
- draw squares at correct locations: 4 points

draw circles:

- draw some circles: 2 points
- draw rows of circles: 2 points
- draw correct number of circles per row: 2 points
- draw correct number of circles in each row at correct location: 5 points