

COP 3330 Quiz #2 2/20/2026 Solutions

Grading Note: Any student responses that call next() or nextInt() into formal parameters or have unnecessary System.out.println will automatically receive a maximum of half credit. (Thus, in these cases, ignore the listed grading criteria and award no more than half credit.)

1) (10 pts) Jamila is making fruit baskets which have both apples and oranges. Each basket will have the same number of apples and the same number of oranges. Sound familiar? This time we'll write a static method that solves the same problem from quiz 1. Your method will take in four formal parameters of type int: aPerBasket, oPerBasket, aTotal and oTotal, representing the number of apples in each basket, the number of oranges in each basket, the total number of apples available for Jamila and the total number of oranges available for Jamila, respectively. Write the numBaskets method so it returns the maximum number of complete baskets she can form. Like last time, for full credit please use a method from the Math library, no if statements and no loops. You may assume that all four formal parameters passed to the method are positive integers.

```
public static int numBaskets(int aPerBasket, int oPerBasket, int aTotal, int
oTotal) {

    return Math.min(aTotal/aPerBasket, oTotal/oPerBasket);

}
```

Grading: 2 pts return, 2 pts Math.min, 3 pts one parameter, 3 pts other parameter, give partial credit on parameters as you see fit, and if statement solution that works should be 8 out of 10, a loop solution that works should be 2 out of 10.

2) (10 pts) Complete the static method below with formal parameters rndObj (Random reference), and size (int), so that it returns an array of size randomly generated integers in between 0 and 100, inclusive.

```
public static int[] getRndInts(Random rndObj, int size) {

    int[] res = new int[size];
    for (int i=0; i<size; i++)
        res[i] = rndObj.nextInt(101);
    return res;

}
```

**Grading: 3 pts array creation (1 pt LHS, 2 pts RHS, 1 pt new 1 pt rest)
2 pts loop (give 1 pt if there's a loop but something's wrong with it)
4 pts assignment (1 pt LHS, 1 pt rndObj, 1 pt .nextInt, 1 pt 101)
1 pt return**

3) (15 pts) Maybe you remember the inverted pyramid from Quiz #1 also. Same deal – write a method that takes in a positive integer n guaranteed to be in between 1 and 9, inclusive, and prints out the corresponding inverted pyramid. To jog your memory, here's an inverted pyramid corresponding to n = 5:

```
5 5 5 5 5
 4 4 4 4
  3 3 3
   2 2
    1
```

Same restrictions as last time. All of your print/println statements can print out at most two characters.

```
public static void printInvertedPyramid(int n) {
    for (int i=n; i>=1; i--) {                // Grading 3 pts
        for (int j=0; j<n-i; j++)            // Grading 5 pts total
            System.out.print(" ");
        for (int j=0; j<i; j++)              // Grading 5 pts total
            System.out.print(i+" ");
        System.out.println();               // Grading 2 pts
    }
}
```

Grading Note: Many ways to do this. Map points as you see fit. Some students last quiz correctly printed spaces before numbers on a line, so trace through those solutions to see if they work.

4) (4 pts) What is the definition I gave in class for the following two words that I asked you all to memorize and that would "eventually make sense":

static – Belongs to the class

instance – Belongs to the object

Grading: 2 pts on each all or nothing. Slight rewording is fine, but the words "class" and "object" must appear in the definition, and whatever their rewording needs to be of similar meaning to "belongs to". I don't care about "the" =>

5) (6 pts) In class, a lot of time was spent explaining why the reduce method in the fraction class was an instance method but the gcd method in the fraction class was a static method. In your own words, explain the reasoning behind this design decision.

The reduce method is an action taken on a fraction which results in a mutation of that fraction object. When I reduce the object 4/8, it becomes the object 1/2. Thus, it makes sense that reduce "belongs to the object" as it is specifically an action to take on a fraction object, similar to read on a Book object. The gcd method is between two integers and has nothing to do with a fraction object per se. It's instead an auxiliary method that helps us with the reduce action, but can be called without regard to a fraction object as well. Thus, it's better to characterize this as a static method that belongs to the class but doesn't belong to a specific fraction object.

Grading: 3 pts for explaining why reduce is instance, 3 pts for explaining why gcd is static, give partial as you see fit.

6) (12 pts) A Point class storing a point on the Cartesian plane and a LineSegment class, storing a line segment between two Point objects is available for you to use. Here is the required documentation from these two classes:

Point class

```
// Creates a Point object with coordinates (x, y).
public Point(double x, double y);
```

```
// Returns the distance from Point this to Point other.
public double distance(Point other);
```

LineSegment class

```
// Creates a LineSegment object between Points p1 and p2.
public LineSegment(Point p1, Point p2);
```

```
// Returns the perpendicular distance from the line defined by
// this LineSegment and the point other.
public double distToPoint(Point other);
```

Using the methods provided above and doing almost no real math on your own (except for multiplying values and dividing by 2), write a static method that takes in three Point objects and returns the area of the triangle defined by those Point objects. You may assume that the Point objects create a triangle with positive area.

```
public static double triArea(Point a, Point b, Point c) {
    LineSegment base = new LineSegment(a, b);
    double height = base.distToPoint(c);
    double baseL = a.distance(b);
    return .5*height*baseL;
}
```

Grading: 3 pts for each line (conceptually as there are different ways to write this with fewer lines of code), give partial as you see fit, but generally, it's very important to call instance methods on objects, so the object. is worth a point always, and each actual parameter must be correct. Note that a.distance(b) could be b.distance(a) and there are a few different ways to do this.

7) (16 pts) Write a static method that takes in a 2D integer array and returns the maximum sum of any row or column. For example, if the array stored the following values:

6	20	13	4	5
1	30	7	9	3
9	15	10	8	2

your method should return 65, the sum of the column at index 1, since this sum is larger than the sums of rows 0, 1, 2 and columns 0, 2, 3, and 4. **You may assume that the array is rectangular meaning that each individual array of integers is the same size. You may also assume that all entries in the array are non-negative. (This makes it easier to set a default result initially.)**

```
public static int maxRowOrColSum(int[][] array) {  
  
    int res = 0; // Grading: 1 pt  
  
    for (int i=0; i<array.length; i++) { // Grading: 1 pt  
        int rSum = 0; // Grading: 1 pt  
        for (int j=0; j<array[i].length; j++) // Grading: 1 pts  
            rSum += array[i][j]; // Grading: 2 pts  
        res = Math.max(res, rSum); // Grading: 2 pts  
    }  
  
    for (int j=0; j<array[0].length; j++) { // Grading: 1 pt  
        int cSum = 0; // Grading: 1 pt  
        for (int i=0; i<array.length; i++) // Grading: 1 pts  
            cSum += array[i][j]; // Grading: 2 pts  
        res = Math.max(res, cSum); // Grading: 2 pts  
    }  
  
    return res; // Grading: 1 pt  
}
```

Grading Note: The nested loops must run different amounts of times. There's a total of 4 pts for the loop iterations. Give partial as you see fit. If both loops have them flipped do 2 of 4, if one loop is flipped but the other correct 3 out of 4, if both loops run the wrong # of times AND not just flipped give no credit for the loop.

8) (2 pts) What Winter Olympic sport involves throwing stones in a trajectory known as a "curl"?

Curling (give to all)

Java Language Reference

Math Class

Modifier and Type	Method	Description
static double	abs(double a)	Returns the absolute value of a double value.
static double	max(double a, double b)	Returns the greater of two double values.
static double	min(double a, double b)	Returns the smaller of two double values.
static double	pow(double a, double b)	Returns the value of the first argument raise to the power of the second argument.

Random Class

Modifier and Type	Method	Description
double	nextDouble()	Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0 from this random number generator's sequence.
int	nextInt(int bound)	Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.