Grade 11 Review Part III – Methods

1. For these questions, make sure you follow conventions for naming.
a) Write a method that takes an `int` as parameter and doubles it. Return the value.
b) Write a method that takes three `double` values and returns the average.
c) Write a method that takes an `int` parameter n and prints a triangle of asterisks starting from 1 to n. Sample output if `n = 5`:

```
*
**
***
****
*****
```

d) Write a method to return the circumference of a circle given the radius as parameter. Use `Math.PI` to get π.

2. Complete the method headers for the provided code. Try to give the method a sensible name. (If you make a copy of this document, you can type it in.)

```
[header] {
    double v = d / t;
    return v;
}
```

```
[header] {
    int fact = 1;
    for (int i = n; i > 1; i--) {
        fact *= i;
    }
    return fact;
}
```

```
[header] {
    if (n%2 == 0) {
        return true;
    }
    return false;
}
```

3. a) This question is related to line segments in the *x-y* plane. Create a project named `Lastname_Firstname_Lines`. Write the following methods in the class:

- `calcSlope`: accepts four doubles as parameters, the coordinates of two points on a line, `x1, y1, x2, y2`, and returns its slope (double). [How to return slopes of horizontal or vertical lines? See Hints below.]

- `calcLength`: accepts four doubles as parameters, the coordinates of two points on the line, `x1, y1, x2, y2`, and returns its length (double)

Hints:

The formula for calculating length of a line segment is $l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

The formula for calculating slope of a line is $m = \frac{y_2 - y_1}{x_2 - x_1}$

Use `Math.sqrt(num)` to find the square root of a number [or `Math.pow(num, 0.5)`]

Horizontal lines have zero slope; vertical lines have infinite/undefined slope; use something like:

```
if (slope == Double.POSITIVE_INFINITY ||
    slope == Double.NEGATIVE_INFINITY) {
   . . .
}
```

or slightly cleaner:

```
Math.abs(slope) == Double.POSITIVE_INFINITY
```

Parallel lines have equal slope; perpendicular lines have negative reciprocal slopes (exception: horizontal and vertical lines).

b) Write a main method which asks the user for the coordinates of four points on two different lines. Then check if the line segments have the same length, and if they are parallel, perpendicular, or neither.

Here is a sample run:

```
Please enter the coordinates of two points on line segment
1:
point A x-value: 0
point A y-value: 0

point B x-value: 1
point B y-value: 1

Please enter the coordinates of two points on line segment
2:
point C x-value: 2
point C y-value: 2
```

```
point D x-value: 4
point D y-value: 4

Results:
The two line segments do not have the same length.
The two line segments are parallel.
```

4. a) Recall the definition of a quadratic equation is of the form $ax^2 + bx + c = 0$. Create a project named `Lastname_Firstname_QuadraticSolver`. Write the following methods:

- `discriminant`: accepts three double values `a, b, c,` and returns the value of the discriminant of the equation, $D = b^2 - 4ac$
- `numSolutions`: accepts three double values `a, b, c,` and returns the number of solutions to the equation
- `solveEquation`: accepts three double values `a, b, c,` and prints the roots, if any, in a nice statement, rounded to two decimal places, if necessary. If there are no real roots, prints a statement to that effect.
- (Optional) `isFactorable`: accepts three double values `a, b, c,` and returns **true** if the equation is factorable over the integers

Hints:
To print a rounded float, use

```
System.out.printf("round to two decimal places: %.2f\n", 23.3467);
// prints “round to two decimal places: 23.35” (with new line)
```

The sign of the discriminant determines the number of solutions (look it up).
The quadratic formula will give the roots of the equation (look it up).
A quadratic expression is factorable if it can be written as the product of two polynomials $(ax - c)(bx - d)$, where *a, b, c* and *d* are integers.

b) Create a main method that prints a nice menu asking the user for the values of `a, b,` and `c,` from their quadratic equation. The program should print the value of the discriminant, the number of solutions, and the numeric roots, if any. (Optional: also print whether the equation is factorable.)