Indiana University
CSCI-C212: Introduction to Software Systems (F25)
Joshua Crotts

Problem Set 1
August 25, 2025

(Basic) Methods, Mathematical Operations

Important Dates:

- Assigned: August 25, 2025
- Deadline: September 3, 2025 at 11:59 PM EST

Objectives:

- Students learn to use basic Java concepts involving different datatypes.
- Students gain experience with the Java mathematics library.
- Students design methods to complete a task and write corresponding unit tests.

What To Do:

For each of the following problems, create a class named `ProblemX`, where `X` is the problem number. E.g., the class for problem 1 should be `Problem1.java`. Write (JUnit) tests for each method that you design in corresponding test files named `ProblemXTest`, where `X` is the problem number. Additionally, write Javadoc comments explaining the purpose of the method, its parameters, and return value. **Do not round your solutions!**

What You Cannot Use:

You cannot use any content outside Chapter 1.1. This includes strings, conditionals, recursion, loops, arrays, regular expressions, data structures, and so forth. You *are* allowed and encouraged to use methods from the `Math` class.

Any use of anything in the above-listed forbidden categories will result in a **zero** (0) on the problem set.

Problem 1:

Design the `double gigametersToLightDays(double gkm)` method, which converts a distance in gigameters to light days (i.e., the distance that light travels in one day). Light travels 299, 792, 458 meters per second, and there are 86,400 seconds in a day.

Problem 2:

Design the `double grocery(int a, int b, int o, int g, int p, int c)` method, which receives six integers representing the number of apples, bananas, oranges, bunches of grapes, pineapples, and coconuts purchased at a store. Use the following table to compute the total purchase cost in US dollars.

Item	Price Per Item
Apple	$1.95
Banana	$2.65
Orange	$3.49
Bunch of Grapes	$1.62
Pineapple	$4.54
Coconut	$3.39

Problem 3:

Design the `double coneSurfaceArea(double r, double h)` method, which computes the surface area of a cone. The formula is:

$$A = \pi r \left(r + \sqrt{(h^2 + r^2)} \right)$$

Problem 4:

The *z-score* is a measure of how far a given data point is away from the mean of a normally-distributed sample. In essence, roughly 68% of data falls between z-scores of $[-1, 1]$, roughly 95% falls between $[-2, 2]$, and 99.7% falls between $[-3, 3]$. This means that extreme outliers have z-scores of either less than -3 or greater than 3.

Design the `boolean isExtremeOutlier(double x, double avg, double stddev)` method that, when given a data point x, a mean μ, and a standard deviation σ, computes the corresponding z-score of x and returns whether it is an "extreme" outlier. Use the following formula:

$$Z = \frac{(x - \mu)}{\sigma}$$

Hint: remember that you cannot use if/switch/conditional statements. How can you leverage a `Math` *class method to solve the problem? You are allowed to use logical and comparison operators, e.g., < and ||.*

Problem 5:

Design the `double lawOfCosines(double a, double b, double th)` method that, when given two side lengths of a triangle a, b and the angle between those two sides θ in degrees, returns the length of the third side c. The formula is listed below. Hint: `Math.cos` receives a value in radians; we convert a value from degrees to radians with `Math.toRadians`.

$$c = \sqrt{a^2 + b^2 - 2ab\cos\theta}$$

Problem 6:

A physics formula for computing object distance displacement is

$$d = t \cdot v_i + (1/2) \cdot at^2$$

where d is the final distance traveled in meters, v_i is the initial velocity, t is the time in seconds, and a is the acceleration in meters per second squared. Design the `double distanceTraveled(double vi, double a, double t)` method that, when given these variables as parameters, returns the distance that the object in question traveled.

Problem 7:

The "square root curve" is a method of scaling grades for an assessment. The idea is to "curve" a student's grade to a "more meaningful" grade. For example, suppose a student scores 65/100 on an exam. Their curved score is $\sqrt{65} \cdot 10 \approx 8.06 \cdot 10 \approx 80.06$. Therefore, the student earned 15.06 points from the square root scale. Design the `double squareRootScalePoints(double score, double maxPoints)` method that, when given a raw score and the maximum number of points for some assessment, returns the number of points earned *from the curve*. For example, `squareRootScalePoints(65, 100)` returns approximately 15.62, because the student earned an 80.06 after the square root scaling. If the exam were out of 80 points, i.e., `squareRootScalePoints(65, 80)`, the method would instead return a value that is approximately 7.111. (Note: this exercise purposefully omits the mathematical details to get you to think about what you're supposed to do!)

Problem 8:

Design the `double billTotal(double t)` method, which computes the total for a bill. The total is the given subtotal t, plus 6.75% of t for the tax, and 20% of the taxed total for the tip.

Problem 9:

Design the `boolean areOddOrSum(int x, int y, int z)` that returns `true` if x or y are odd, but not both (and not neither), OR if the sum of x and y is equal to z.

Remember, again, you can't use `if` statements or similar. You can use comparison operators such as == and logical connectives `||`.

Problem 10:

Design the `double angle(double a, double b, double c)` method that, when given three side lengths of a triangle a, b, and c, returns the angle θ opposite to that of c in degrees. The formula for this computation is listed below. Hint: arccosine in Java is `Math.acos`.

$$\theta = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$