

## Instructions

Please read the instructions carefully and follow the naming conventions specified for each question. Solutions must be submitted in the Blackboard Dropbox created for Lab 1.

The deliverable will be solutions to the questions below (placed a single package named `btp400.lab1`). The submission shall be in a single jar file (called `btp400lab1.jar`) which contains both source (\*.java) and bytecode (\*.class) files. Your solution should be well documented using the JavaDoc utility to describe both your interface and your solution design.

Note that the deadline is strictly enforced. The system tracks the exact time that submissions are uploaded. **There is a 10% per day penalty for late submissions.**

## Additional Notes

- You must provide appropriate test case classes with your solution. Document any assumptions you make about the requirements. You will have to organize your code in a way such that the JUnit test runner does not require user input. JUnit 5 must be used to create your test harness and classes
- You will be required to present and explain your solution to the professor during the lab period.
- You may use any IDE for development but note that demonstrations and professor testing will be done exclusively on the command line.

## Question Descriptions

**Question 1)** Write a class called `Pascal` that stores a Pascal's triangle for a given `n` using the generic class, `ArrayList<ArrayList<Integer>>`  
<https://www.mathsisfun.com/pascals-triangle.html>

**Question 2)** Write a class called `SquareMatrix` that accepts a two-dimensional array of integers as an argument to its constructor. A method called `isSquare()` should return true if it is a magic square. The array is "magic" if the sum of all rows, columns and diagonals is the same.

**Question 3)** Write a class called `ISBN` which has two static methods defined as follows

- (a) `buildISBN`, which accepts the first 9 digits of an ISBN number and returns its associated valid ISBN number
- (b) `verifyISBN`, which accepts an ISBN number and returns if the number is valid or not

NOTE: a valid ISBN number contains 10 digits. The first 9 digits are arbitrary, decimal digits (i.e. range from 0..9). The 10<sup>th</sup> digit is alphanumeric and a function of the first nine.

If  $d_0 \dots d_9$  represent the ordered digits, then the last digit  $d_9$  is defined by evaluating the formula below.

$$(d_0 + 2d_1 + 3d_2 + 4d_3 + 5d_4 + 6d_5 + 7d_6 + 8d_7 + 9d_8) \% 11,$$

If the formula evaluates to  $< 10$ ,  $d_9$  is set using the result of the evaluation, otherwise  $d_9$  should be set to the character digit "X".