# Tutorial/Lab 2 – OOP Review 2

## Aim

This tutorial/lab aims to review object-oriented programming concepts, such as abstraction, encapsulation, inheritance and polymorphism. ArrayList is also reviewed in this session.

## Tips

Create a good set of test cases first, which can cover all kinds of cases of input values. This will be helpful to test your code later after you write your solution, but also in understanding the problem statement before you start solving the problem.

## Exercise 2.1 My 2D Point Implementations

- Design a class named `MyPoint` to represent a point with x- and y-coordinates.
- The class contains:
  - The data fields (instance variables) x and y that represent the coordinates with getter methods.
  - An empty constructor that creates a point (0.0, 0.0).
  - A constructor that constructs a point with specified coordinates.
  - A (instance) method named `distance` that returns the distance from this point to another point of the `MyPoint` type.
  - A static method (function) also named `distance` that returns the distance from two `MyPoint` objects.
- Write a test program that creates the three points (0.0, 0.0), (10.25, 20.8) and (13.25, 24.8) and displays the distance between them using both distance implementations.
  Notice the difference between invoking an instance method and a static method (functions)!

## Exercise 2.2 Big Integers Divisible by 2 or 3

- Write a function that prints the first 10 numbers with 50 decimal digits that are divisible by 2 or 3.

## Exercise 2.3 Person, Student, Employee, Faculty, Staff

- Design a class named `Person` and its two **subclasses** named `Student` and `Employee`.
- Make `Faculty` and `Staff` **subclasses** of `Employee`.
- A person has a name, address, phone number, and e-mail address.
- A student has a class status (freshman, sophomore, junior, or senior).
  Define the status as a constant.
- An employee has an office, salary, and date hired.
  Use the `LocalDate` class to create an object for date hired.
- A faculty member has office hours and a rank.
- A staff member has a title.
- Add a constructor that only takes name to each class (you may add other constructors).
- Override the `toString` method in each class to display the class name and the person's name.
- What should the access modifier of `name` in `Person` be?
- Write a test program that creates a `Person`, `Student`, `Employee`, `Faculty`, and `Staff` objects in an array, and invokes their `toString()` methods using **polymorphism**.

## Exercise 2.4 MyStack using Inheritance

- In Lecture 2, we implement `MyStack` using composition.
- Now, implement a new `MyStack` class that extends `ArrayList` Instead.
  - The class definition should be
    `public class MyStack extends ArrayList<Object>`
  - The methods are `isEmpty()`, `getSize()`, `peek()`, `pop()`, `push(Object o)`, `search(Object o)` and `toString()`.
- Write a test program that prompts the user to enter five strings and displays them in reverse order.

This is the end of CPT204-2425 Tutorial/Lab 2 Task Sheet.