# Chapter 2: Class and Object in Python with GIS Application Emphasis

## Overview

The two exercises aim to enhance students' understanding of object-oriented programming and spatial analysis using Python. In Exercise 1, students will extend the `GISPoint` class to include a method that calculates the distance to another point, thereby gaining practical experience in method implementation and class extension. Exercise 2 requires students to create a `GISLine` class, representing a line composed of multiple `GISPoint` objects, and calculate the total length of this line. This will help students understand how to manage collections of objects and apply their methods in practical scenarios.

## Due date

The exercises should be completed by the 20th of November and the code submitted in a folder named after the student via Telegram.

## Points: 4%

## Form of exercise:

Individual

## Exercise 1: Extend the GISPoint Class

### Objective:

Extend the GISPoint class to include a method that calculates the distance to another point. Refer to the lecture and practical material of chapter 2 of this course.

### Steps:

* Define the GISPoint class with attributes for the coordinates, such as x and y.
* Implement a method distance\_to within the GISPoint class that takes another GISPoint object as a parameter.
* Use the Euclidean distance formula to calculate the distance between the two points.

math.sqrt((self.x - other\_point.x)\*\*2 + (self.y - other\_point.y)\*\*2)

### Explanation:

The distance\_to method calculates the distance between the current point and another point using the Euclidean distance formula:

## Exercise 2: Create a GISLine Class

### Objective:

Create a GISLine class that represents a line made up of multiple GISPoint objects.

### Steps:

Define the GISLine class with an attribute for storing multiple GISPoint objects.

Implement methods to add points to the line and calculate the total length of the line by summing the distances between consecutive points.

### Explanation:

* The GISLine class holds a list of GISPoint objects.
* The add\_point method allows adding new points to the line.
* The total\_length method calculates the total length of the line by summing the distances between consecutive points using the distance\_to method from the GISPoint class.