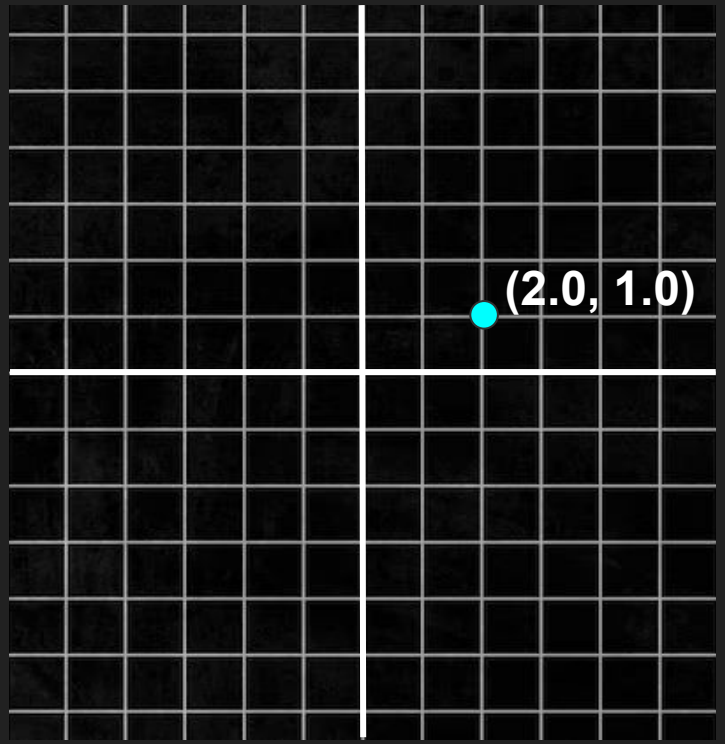
CL28: OOP Practice

Let’s use some Point objects to make a Line!

Consider this Point class

```
0 class Point:
1     x: float
2     y: float
3
4     def __init__(self, x: float, y: float):
5         self.x = x
6         self.y = y
7
8     def dist_from_origin(self) -> float:
9         return (self.x**2 + self.y**2) ** 0.5
10
11    def translate_x(self, dx: float) -> None:
12        self.x += dx
13
14    def translate_y(self, dy: float) -> None:
15        self.y += dy
16
17 pt: Point = Point(2.0, 1.0)
```



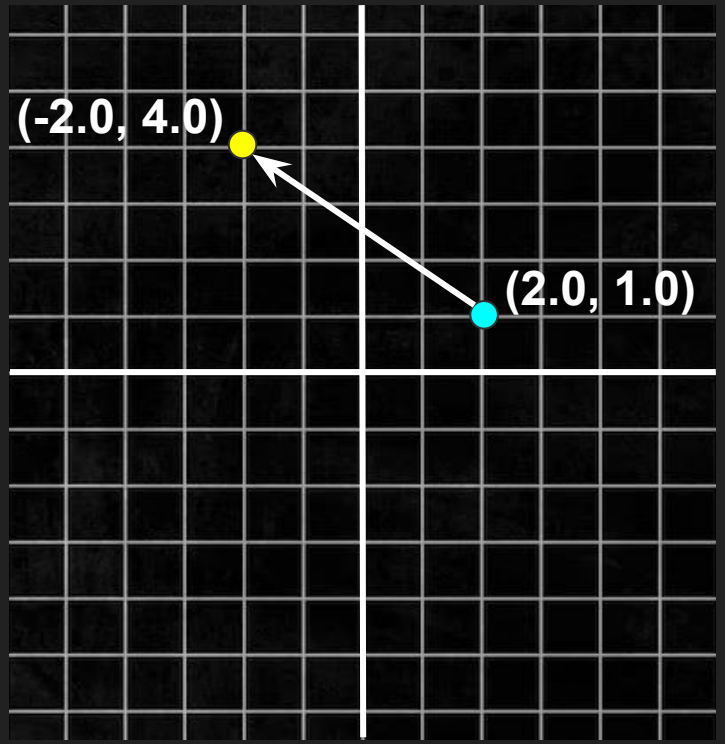
“Two points make a line”

Finding the length of a line:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Finding the slope of a line:

$$m = \frac{\text{Rise}}{\text{Run}} = \frac{y_2 - y_1}{x_2 - x_1}$$

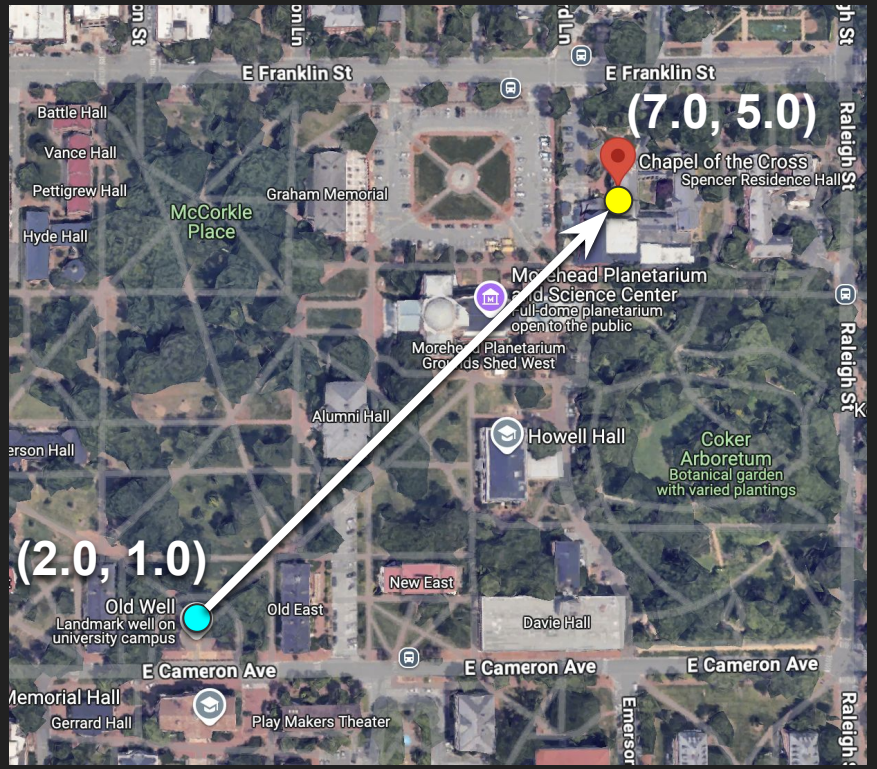


“Two points make a line”

Let’s make a Line class and use it to see the distance from the Old Well to the Planetarium!

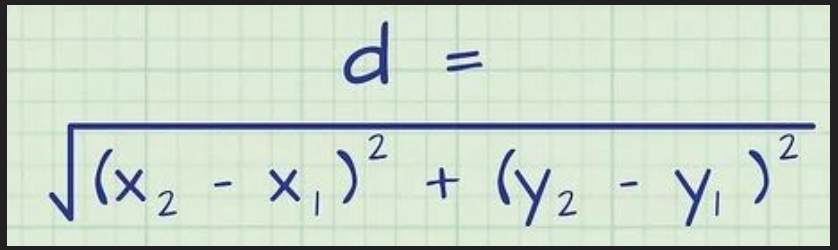
On paper, write a **`Line`** class with two attributes: a starting point (**`start: Point`**) and an ending point (**`end: Point`**).
The **`Line`** class should have the following method definitions:

- `def __init__(self, start: Point, end: Point):`
- `def get_length(self) -> float:`
  - Calculates the length of the line
- `def get_slope(self) -> float:`
  - Calculates the slope (from **`start`** to **`end`**)

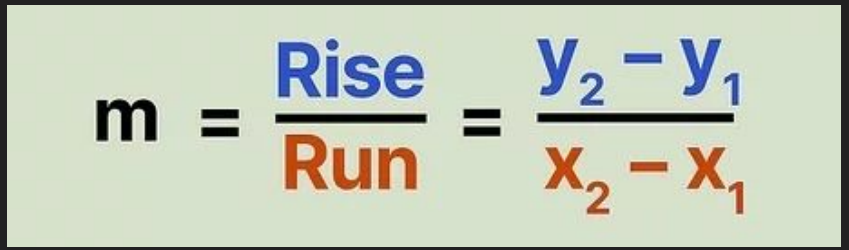


“Two points make a line” – Let’s make a Line class!

Finding the length of a line:



Finding the slope of a line:



On paper, write a `Line` class with two attributes: a starting point (`start: Point`) and an ending point (`end: Point`).
The `Line` class should have the following method definitions:

- `def __init__(self, start: Point, end: Point):`
- `def get_length(self) -> float:` calculates the length of the line
- `def get_slope(self) -> float:` calculates the slope (from `start` to `end`)

Let’s go over it together! →

Submit a .pdf of your hand-written code to Gradescope!