

# Assignment - Classes

## 1. Basic Class

### 1.1 Class Point

Create a Python class `Point` which contains 2 attributes, `x` and `y`, representing `x` and `y` coordinate of the point.

- Implement initializer method which initialize `x` and `y`.
- Implement a instance method `dist_to_origin()` which return distance from origin using formular  $\text{math.sqrt}(x^2 + y^2)$ .
- Implement `__str__()` which returns "`(x,y)`", e.g. "`(3.0,4.0)`"

#### Sample Output

```
(3.0, 4.0)
5.0
```

```
In [ ]: 1 class Point:
        2
        3     def __init__(self, x = 1, y = 1):
        4         self.x = x
        5         self.y = y
        6
        7     def dist_to_origin(self):
        8         import math
        9         return math.sqrt(self.x**2 + self.y**2)
       10
       11     def __str__(self):
       12         return '({},{})'.format(self.x, self.y)
       13
       14 p = Point(3, 4)
       15 print(p)
       16 d = p.dist_to_origin()
       17 print(d)
       18
```

### 1.2 Class Rectangle

Create a Python class `Rectangle` which contains 3 attributes, `width`, `height` and `corner`. The `corner` is of `Point` type, which gives coordinate of bottom left corner of the rectangle.

- Implement initializer method which initialize `width`, `height` and `corner`.
- Implement a instance method `get_centre()` which returns a `Point` object representing centre point of the rectangle.

- Implement a instance method `scale(val)` which scale width and height by `val` times.

### Sample Output

```
(12.0, 22.0) # print return value from `get_centre()`  
Rectangle(20, 40) at point (2.0, 2.0) # return value of `str()`
```

```
In [ ]: ▶ 1 class Rectangle:
2
3     def __init__(self, w, h, p = Point(0,0)):
4         self.width = w
5         self.height = h
6         self.corner = p
7
8     def get_centre(self):
9         x = self.width / 2 + self.corner.x
10        y = self.height / 2 + self.corner.y
11        return Point(x, y)
12
13    def scale(self, val):
14        self.width = self.width * val
15        self.height = self.height * val
16
17    def __str__(self):
18        return 'Rectangle({}, {}) at point {}'.format(self.width, self.h
19
20 r = Rectangle(20, 40, Point(2,2))
21 print(r.get_centre().dist_to_origin())
22
23
```