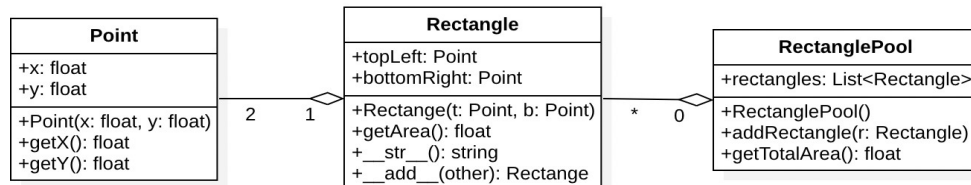




## Object Oriented Design and Programming with Python.

### 1. Implement the following class diagram in Python



#### Point Class

##### Attributes

- x: the x coordinate of a point
- y: the y coordinate of a point

##### Methods:

- getX(): returns the x coordinate
- getY(): returns the y coordinate

#### Rectangle Class

##### Attributes

- topLeft: the top left point of a rectangle
- bottomRight: the bottom right point of a rectangle

##### Methods

- getArea(): returns the area of a rectangle — width x height
- \_\_str\_\_(): returns the string representation of a rectangle in the following form: [(topleft-x, topleft-y) (bottomright-x, bottomright-y)]
- \_\_add\_\_(): creates and returns a new rectangle by adding the x and y coordinates of the points of one rectangle to the others.

#### RectanglePool Class

##### Attributes

- Rectangles: the list of rectangles

##### Methods

- addRectangle: add a given rectangle to the list of rectangles
- getTotalArea(): returns the total area of all the rectangles in the list

### 2. Create a class TicTacToe that will enable you to write a complete program to play the game of tic-tac-toe. The class contains a 3-by-3 double-subscripted list of letters. The constructor should initialize the empty board to all zeros. Allow two human players. Wherever the first player moves, place an "X" in the specified square; place an "O" wherever the second player moves. Each move must be to an

empty square. After each move, determine whether the game has been won and whether the game is a draw.

Modify your program so that the computer makes the moves for one of the players automatically.

3. Create a class called Complex for performing arithmetic with complex numbers. Write a driver program to test your class.

Complex numbers have the form  $\text{realPart} + \text{imaginaryPart} * i$

where  $i$  is

$$\sqrt{-1}$$

Use floating-point numbers to represent the data of the class. Provide a constructor that enables an object of this class to be initialized when it is created. The constructor should contain default values in case no initializers are provided. Provide methods for each of the following:

- a. Adding two ComplexNumbers: The real parts are added to form the real part of the result, and the imaginary parts are added to form the imaginary part of the result.
- b. Subtracting two ComplexNumbers: The real part of the right operand is subtracted from the real part of the left operand to form the real part of the result, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand to form the imaginary part of the result.
- c. Enabling output of complex numbers in the form  $(\text{realPart}, \text{imaginaryPart})$ , through the overloaded `_str` method.

#### References:

Deitel, Harvey M., Deitel Paul J., Liperi, Jonathan P. and Wiedermann, Ben (2002) *Python How to Program*. Prentice Hall. ISBN: 0130923613