

## Today

- Expression evaluation revisited
  - Precedence & Associativity Together
  - Fully parenthesized expressions
- Object Creation Review
- Object Creation Testing
- Quiz Today

$$(2 + (4 * 9))$$

precedence  
 $*$  higher (compute earlier)  
 $+$  lower

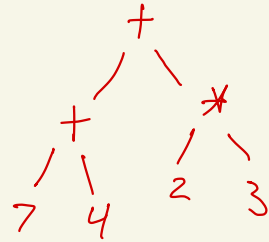
associativity

$$((2 + 4) + 9)$$

$$((2 - 4) - 9)$$

left-associative

$$((7 + 4) + (2 * 3))$$



$$\boxed{2 + 4} < 6 * 10$$

$$6 < 6 * 10$$

$$6 < 60$$

# Objects

```
class Point :  
    def __init__(self, x_coord : float, y_coord : float) :  
        self.x = x_coord  
        self.y = y_coord
```

↑  
attribute names  
w/in object

xvalue = 9.2

yvalue = 4.7

p1 = Point(xvalue, yvalue)

xvalue = 9.2

yvalue = 4.7

p1 = Point(xvalue, yvalue) ←

~~p1 = Point(9.2, yvalue)~~

~~p1 = Point(9.2, 4.7)~~ ← (results in evaluates to the new object)

def \_\_init\_\_(self, x\_coord: float, y\_coord: float):

self.x = x\_coord (9.2)

self.y = y\_coord (4.7)

Known w/in  
\_\_init\_\_

x\_coord [9.2]

y\_coord [4.7]

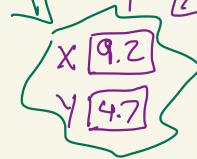
self [ ]

Known

xvalue [9.2]

yvalue [4.7]

p1 [ ]



- values w/in object are named
- names are called

attributes

properties

fields

(data / state)

data.py

```
class Point:  
    :
```

data\_tests.py

```
class PointTests(unittest.TestCase):
```

```
    def test_point(self):
```

```
        result = Point(2.6, 4.2)
```

```
        self.assertEqual(result.x, 2.6)
```

```
        self.assertEqual(result.y, 4.2)
```

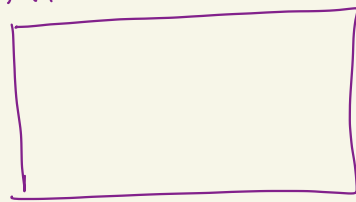
```
    def test_point2(self):
```

```
        final_answer = Point(9.1, 2.7)
```

```
        self.assertEqual(  
            final_answer.x, 9.1)
```

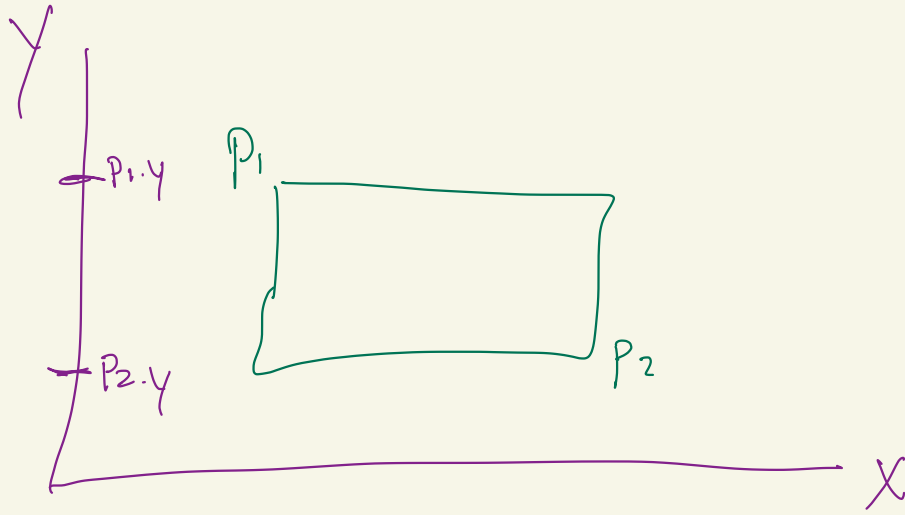
```
        self.assertEqual(  
            final_answer.y, 2.7)
```

$x_1, y_1$



$x_2, y_2$

$$(x_2 - x_1) * (y_1 - y_2)$$



$p1 = \text{Point}(\sim, \sim)$

$p2 = \text{Point}(\sim, \sim)$

$$(p2.x - p1.x) * (p1.y - p2.y)$$



```
def rectangle_area(p1: Point, p2: Point):  
    return (p2.x - p1.x) * (p1.y - p2.y)
```