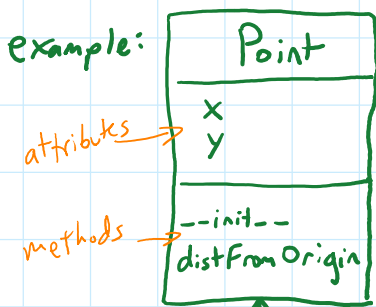
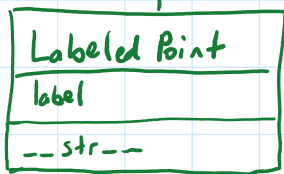


Inheritance: one class can inherit attributes and methods from another



Parent

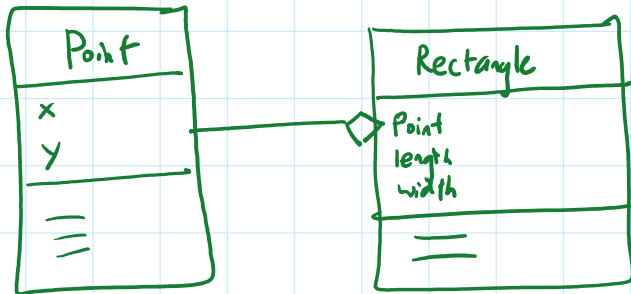
INHERITANCE  
LabeledPoint is a Point



Child

← this class can utilize all of its own attributes/methods and those from Point

~~Rectangle is a Point~~



COMPOSITION  
Rectangle has a Point

EXAMPLE: Polynomials

$$x^2 + 5x - 3$$

$$3x^4 - 2x + 7$$

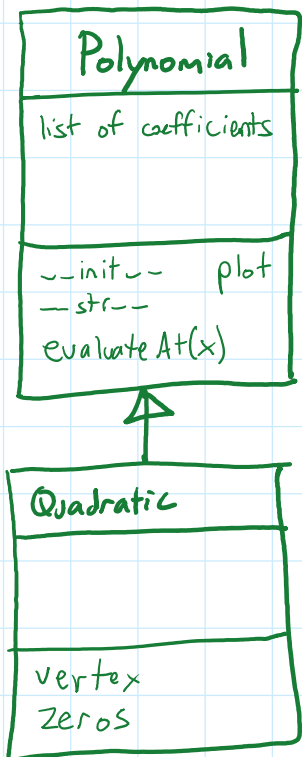
general polynomial:  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$

↑ coefficients

quadratic:  $a_2 x^2 + a_1 x + a_0$

A quadratic is a polynomial.

A quadratic is a polynomial.



class Polynomial():

def \_\_init\_\_(self, coeffList):

≡

def \_\_str\_\_(self):

return \_\_\_\_\_

$[a_0, a_1, a_2, \dots, a_n]$   
element of the list at index  $i$  will be the coefficient of  $x^i$

class Quadratic(Polynomial):

def \_\_init\_\_(self, a, b, c):

quadratic  $ax^2 + bx + c$

---

```
def evaluateAt(self, x):
    s = 0
```

```
    for i in range(len(self.coeff)):
```

```
        s += c * (x ** i)
```

self.coeff[i]

---

Nice solution:

```
s = 0
```

```
for i, c in enumerate(self.coeff):
```

index value from the list

```
s += c * (x ** i)
```

HOMEWORK:

