Example of
Object Oriented Programming
In
Python Programming Language

Class

```
class Person(object):
    def __init__(self, name):
        self.name = name
```

Object

```
p1 = Person('A')
print(p1.name) #A

p2 = Person('B')
print(p2.name) #B

print(id(p1)) #memory address of this object
```

No real private access control

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def __private(self):
        print('private')

p1 = Person('A')

p1.__private() #AttributeError: 'Person' object has no attribute '__private'

p1._Person__private()
```

Inheritance

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def say_hello(self):
        print('Hello from %s' %(self.name))


class Student(Person):
    pass


p1 = Person('A')
p1.say_hello()  #Hello from A

p2 = Person('B')
p2.say_hello()  #Hello from B

s1 = Student('C')
s1.say_hello()  #Hello from C
```

Polymorphism: abstract

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def spend_time(self):
        raise NotImplementedError


p1 = Person('A')
p1.spend_time() #NotImplementedError
```

Polymorphism: subclass

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def spend_time(self):
        raise NotImplementedError


class Student(Person):
    def spend_time(self):
        print('Studying')


class Teacher(Person):
    def spend_time(self):
        print('Teaching')

s1 = Student('A')
s1.spend_time() #Studying

t1 = Teacher('B')
t1.spend_time() #Teaching
```

Operator overloading (1)

```
class Person(object):
    def __init__(self, name):
        self.name = name


print(1+1) #2

p1 = Person('A')
print(p1+p1) #TypeError: unsupported operand type(s) for +: 'Person' and 'Person'
```

Operator overloading (2)

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def __add__(self, x):
        return Person(self.name + x.name)

p1 = Person('A')
p2 = p1 + p1
print(p2.name) #AA
```

More information

- https://docs.python.org/3/reference/datamodel.html