

SI 506 DYU 7

The Problem Set this week really tested my understanding of all the previous concepts. I had trouble understanding a few concepts because I had a bad experience learning Java and my understanding of Object Oriented Programming was not clear. I went through the official Python documentation, and my concepts are now clear. I did not do the lecture prep thoroughly because I felt I might jumble up things again. So here is my understanding of Python's OOP aspect.

- Object is an instance of a class
- Method is a function belonging to an object
- Two ways of using objects
- Attribute references (ClassName.attribute) — (A)
- Instantiation (Create an empty object) — (B)

```
class MyClass:
    def function(self):
        return "This is a function"
--
MyClass.function(Function object) (A)
```

```
x=MyClass()      #Create an instance of the class
x.function()      (Method object) (B)
```

Here is something I came up with, to understand the concept of classes, objects, and inheritance:

Consider a restaurant serving varieties of pizzas. At the restaurant, you can choose from available pizza types such as Pepperoni, Margherita, Neapolitan, New York style, Chicago style, etc. According to the type of pizza, you have crust options such as deep dish, thin crust or regular crust pizza. For every pizza type, you have options in toppings such as olives, jalapeños, tomatoes, onions, mushroom, corn, any type of meat and a layer of any cheese you want.

Pizza is a class here. And each type of pizza is an instance of the class Pizza. As with any class, there are some attributes. Each pizza type has certain toppings and crust combination i.e. it has some attributes. Type of crust is one attribute. Every topping is an attribute itself. So, a Neapolitan pizza is an instance of the class Pizza, that is it has attributes such as crust, toppings defined in the class Pizza. The instance of a class is an object.

```
class Pizza:
    crust="regular"
    def __init__(self, toppings,cheese):

        self.toppings=toppings
        self.cheese=cheese

    def ready(self):
        return "Here is what you ordered - "

    def base(self):
        return "A {} crust pizza".format(self.crust)

#Place your order
topping=['mushrooms', 'tomatoes', 'onions', 'olives', 'jalapenoes']

custom=Pizza(topping,"parmesan")

#Confirming the order
print custom.ready()
print custom.base()
print "Topped with {} and gooey melted {} cheese on
top".format(custom.toppings,custom.cheese)
```

This code works, but unfortunately it does not output a pizza!