**Python Day 2 – OOP**

1. What does OOP stand for? What is it, briefly?

* Object Oriented Programming.
* Style of programming uses a blueprint for modeling data.
* Blueprint defined by the programmer – DIY datatypes – customize it for your situation.
* Modularizing

1. **What is a class?**

* User/Programmer defined datatype.
* Like a function is a recipe – class is blueprint for the datatype.

1. **Classes contain 2 things:** *attributes* and *methods*. Give a brief explanation for each:  
      
   ***Attributes:***Characteristics shared by all instances of the class type. Variables that belong to the class.

* What a class of objects have --- datatypes #ex: car
* --model—string (corolla)
* --make—string (Toyota)

***Methods*:** Actions that an object can perform.

* A user, for example, should be able to make a deposit or a withdrawal, or maybe send money to another user.
* Functions that affect the properties of the class.
* Function that belongs to the class.
* What a class of objects can ***do*** – actions/functions.

1. The following are lines of code could belong to a Shopping Cart class but are out of order but. On the right, arrange the code to make a functional class.  
      
   *Once you re-order the code on the right, put a \* next to any attributes, and a box around any methods.*

*Class ShoppingCart:*

self.store = store

self.items.append(item)

def

def add\_item(self, item, price):

self.items = append item []

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

class ShoppingCart:

return self

self.total = 0

self.total += price