Classes and Objects

Inheritance

In programming it is helpful to compartmentalize (or group together) a set of functions, data, or attributes. This practice is helpful when testing/debugging code, you can find out which sections are faulty this way.

This style of coding is called Object Oriented Programming: a style of programming that allows you to model real world concepts in order to create complex programs.

Classes: a class is an abstract data structure that lets you define functions that are special for a certain abstract idea.

* A class is a user defined data structure that contains information about your ‘class’
* Classes are used to create objects

Objects: an object is an instance (or entity/item) of class

* Objects are a special data type, unlike strings or integers if you try to print an object it will print a memory location instead of the value of your object (we will do examples)

Object Attributes: when you create an object, you can give it specific values

* A class for a dog might have values of name, or color, or age value

def \_\_init\_\_() : a reserved method in python classes that is called a class “constructor”. This method is called when an object is created from the class – it defines what attributes an object has.

def \_\_str\_\_(): is a method that will allow you to print the values of an object rather than the memory address.

* For this we can use string formatting:

return “{0:.2f}”.format(thing to print)

Inheritance: When creating classes using Object Oriented Programming you create a hierarchy of classes

* Classes can inherit methods (aka functions) from other classes
* The Parent class (in our example it is the food class) is often a basic level of attributes for your hierarchy of classes and so each child class will inherit the attributes of the parent class
  + Tomato class – a child class of food class – inherits the price attribute

Ex:

Let’s say you are a grocery store and you want to create an online ordering system for your website. Your store sells many different types of food, each item is different but overall there are things they have in common.

Things in Common:

Each item regardless of what it is has a price,

Things not in Common:

Type of item: fruit, meat, drinks, desserts

So, for this example we would create a general food class, and we could also create a fruit, meat, drinks, or desserts classes.

Homework to Try:

1. Use this method Point2d (x,y) coordinates on a 2d coordinate plane, and define a method in the class so that you can find the distance between any two points (and not just distance of a point from the origin.

class Point2d:

def \_\_init\_\_(self, myX, myY):

self.x = myX

self.y = myY

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

return "({},{})".format(self.x, self.y)

def dist(self):

return math.sqrt(self.x\*\*2+self.y\*\*2)

def point\_dist(x, y):

write this method

1. Create a class called bank account and create an initialize method, then a deposit(), withdraw
   1. Each method will just show adding money to the account (deposit) or subtracting money (withdraw)
   2. Write a main method that creates a bank account object and then tests your methods