**Maths.py**

Here is an example of how object orientation might be used in the real world. What we will do is make our own little maths module borrowing some functions we made several weeks ago.

First of all we will define our class and its constructor.

class maths:

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

self.pi = 3.14159

This class doesn’t need much aside from the constant ‘pi’. Now we can make a couple of methods for this class. To begin with we’ll make our own addition method.

def add(self, num\_a, num\_b):

return num\_a + num\_b

This is pretty simple so we’ll add something a bit more specific. We’ll add two methods to calculate a rectangle’s area and perimeter.

def r\_area(self, width, height):

return width \* height

def r\_perimeter(self, width, height):

return (width \* 2) + (height \* 2)

After those we’ll add another two for calculating the area and circumference of a circle.

def circ(self, radius):

return int(2 \* self.pi \* radius)

def circle\_area(self, radius):

return int(self.pi \* (radius \*\* 2))

Finally, we’ll create a method that will tell us if a number is even or not.

def is\_even(self, number):

if(number % 2 == 0):

return True

else:

return False

With that we can create our own maths object and invoke its methods for various mathematical purposes.

m = maths()

print("18 + 83 = {}".format(m.add(18, 83)))

print("Area of 16x43 rectangle: {}".format(m.r\_area(16, 43)))

print("Area of circle with radius 210: {}".format(m.circle\_area(210)))

for x in range(20):

if(m.is\_even(x)):

print(x)

You might consider adding your own methods to make the class more complete. Hopefully this gives you a better idea of how object orientation might be used in the real world, where like functions are bundled up together into their own object, rather than having functions thrown about all over the place.