Python Advance – 3

1. **What is the concept of an abstract superclass?**

An abstract class can be considered as **a blueprint for other classes**. It allows you to create a set of methods that must be created within any child classes built from the abstract class. A class which contains one or more abstract methods is called an abstract class.

1. **What happens when a class statement’s top level contains a basic assignment statement?**

An assignment statement evaluates the expression list (remember that this can be a single expression or a comma-separated list, the latter yielding a tuple) and assigns the single resulting object to each of the target lists, from left to right.

1. ) Why does a class need to manually call a superclass&#39;s \_\_init\_\_ method?

The main reason for always calling base class \_init\_\_ is that **base class may typically create member variable and initialize them to defaults**. So if you don't call base class init, none of that code would be executed and you would end up with base class that has no member variables

4) How can you augment, instead of completely replacing, an inherited method?

class Manager(Person):

def giveRaise(self, percent, bonus=.10):

Person.giveRaise(self, percent + bonus)

This code leverages the fact that a class method can always be called either through an instance (the usual way, where Python sends the instance to the self argument automatically) or through the class (the less common scheme, where you must pass the instance manually). In more symbolic terms, recall that a normal method call of this form:

instance.method(args...) is automatically translated by Python into this equivalent form:

class.method(instance, args...)

where the class containing the method to be run is determined by the inheritance search rule applied to the method's name. You can code either form in your script, but there is a slight asymmetry between the two—you must remember to pass along the instance manually if you call through the class directly. The method always needs a subject instance one way or another, and Python provides it automatically only for calls made through an instance. For calls through the class name, you need to send an instance to self yourself; for code inside a method like giveRaise, self already is the subject of the call, and hence the instance to pass along.

5.) How is the local scope of a class different from that of a function?

Local Scope **occurs when you create a variable inside a function**. By doing that, the visibility and accessibility of the variable is only allowed within that function. Any variable created inside the yellow box is a local variable, just like any variable inside blue box is a global one.