1. What is the concept of an abstract superclass?

Abstract superclass is the one which don’t have any body or only function declarations but don’t have the definitions.

The class which is inheriting has to implement its methods.

Abstract class methods are abstract.

2. What happens when a class statement's top level contains a basic assignment statement?

Then this will be treated as a class variable and can be accessed with the help of class name outside the class.

This is also treated as a default value if it’s getting called into a function.

3. Why does a class need to manually call a superclass's \_\_init\_\_ method?

If it doesn’t call the super class methods manually then it will not be able initialize the super class variable through init method.

As in python it will not implicitly call the super class init method so we need to call the same as below:

super()\_\_init\_\_()

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

Augmenting a method means adding some addition functionality on top of the existing super class method.

We can do that by keeping the same contents of the superclass method and on top of that we can add some additional code into it.

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

The attributes defined in the class body can be used in the functions and in class.

Whereas the attributes defined in the functions are having the scope limited to the function body, they can’t be used outside the function.

Also we can access the class variable with the help of class-names outside the class which is not the case with the functions.