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

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

Once modu.py is found, the Python interpreter will execute the module in an isolated scope. Any top-level statement in modu.py will be executed, including other imports if any. Function and class definitions are stored in the module’s dictionary. Its not just imports and declarations, but any code that occurs outside functions and classes.

Example

Import sys #top level

X=4. #top level

def find(): #top level

return 2

If X:

print 0

else:

print 2

Class finded(object):

X=0

def f():

print 5

In python, the statements which are not indented are called a top-level statement. Internally python gives a special name to top-level statement s as \_\_main\_\_.

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

\_\_init\_\_() of the superclass ( ) will be called automatically. super() returns a delegate object to a parent class, so you call the method you want directly on it..

super() to call the \_\_init\_\_() of the parent class, allowing you to use it in the Subclass without repeating the code.

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

A subclass can either completely override the implementation for an inherited method or the subclass can enhance the method by adding functionality to it.Replacing a Superclass's Method Implementation. Sometimes, a subclass will want to replace entirely its superclass's implementation of a method. Indeed, many superclasses provide an empty method implementation with the expectation that most, if not all, subclasses will completely replace the superclass's implementation of the method

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

The scope of a local class is limited only within the function definition. The function can use the class as usual as local variables. The class gets destroyed as soon as the function is returned.