1. What is the concept of an abstract superclass?

Ans:

A class is called an Abstract class if it contains one or more abstract methods. An abstract method is a method that is declared, but contains no implementation. Abstract classes may not be instantiated, and their abstract methods must be implemented by their subclasses.

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

Ans:

When a Class statement's top level contains a basic assignment statement, it's usually treated as a class attribute or class level variable.

whereas assignment statements inside methods are treated as instance attributes or local attributes.

When an instance of a class is created a single copy of class attributes is maintained and shared to all instances of class. whereas each instance object maintains its own copy of instance variables.

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

Ans:

The main reason for always calling base class \_init\_\_ is that base class may typically create member variables 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 a base class that has no member variables.

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

Ans: Message forwarding allows you to augment an inherited method in such a way that it can perform its inherited action and some new action.

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

Ans:

A Variable which is defined inside a function is local to that function. it is accessible from the point at which it is defined until the end of the function, and exists for as long as the function is existing.

Similarly a variable inside of a class also has a local variable scope. Variables which are defined in the class body (but outside all methods) are called class level variables or class attributes. they can be referenced by there bare names within the same scope, but they can also be accessed from outside this scope if we use the attribute access operator (.). on a class or an instance of the class.