1. What is the concept of an abstract superclass?

Ans: 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. An abstract method is a method that has a declaration but does not have an implementation. While we are designing large functional units we use an abstract class. When we want to provide a common interface for different implementations of a component, we use an abstract class.

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 is identified as class attribute by the particular class and consider it as a constant value for the further operations in the methods of that class.

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

Ans: Class need to manually call a superclass’s \_\_init\_\_ method to inherit the \_\_init\_\_ function of the super class.

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

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

Ans: A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.

A function is the one which uses this local scope and returns some output with the help of logic written inside the function.