1. What is the concept of an abstract superclass?

Ans : In object-oriented programming, an abstract superclass is a class that is designed to be inherited by other classes. It contains abstract methods and variables that are not implemented in the superclass, but instead are defined in its subclasses. The purpose of an abstract superclass is to define a common set of attributes and behaviors that can be shared by all its subclasses, while still allowing for variations and specializations in the behavior of each subclass.

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

When a class statement's top level contains a basic assignment statement, the assignment statement is executed when the class is defined, and the resulting value is stored as a class attribute.

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

The reason for this is that the superclass's \_\_init\_\_ method may contain important initialization code that the subclass needs to run in order to properly initialize its own attributes and methods. If the subclass does not call the superclass's \_\_init\_\_ method, the initialization code in the superclass may not get executed, which can lead to unexpected behavior.

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

Method overriding is a feature in which a subclass provides its own implementation of a method that is already defined in its superclass. To override a method, the subclass must define a method with the same name and parameters as the method in the superclass. To augment, or modify, the behavior of the method, the subclass can call the superclass's implementation of the method using the super() function. The super() function returns a temporary object of the superclass, which allows you to call its methods.

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

Classes and functions define local namespaces, the local scope of a class is designed to support object-oriented programming concepts such as instance variables, class variables, methods, and inheritance, while the local scope of a function is designed to support the encapsulation of code and data.