1. What is the concept of an abstract superclass?

The concept of an abstract superclass in object-oriented programming refers to a class that is designed to be inherited from, but cannot be instantiated directly. It serves as a blueprint or template for other classes, defining common attributes and behaviors that its subclasses can inherit and implement

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, it creates a class attribute. Class attributes are variables that are defined within the class body, but outside of any class methods. These attributes are shared among all instances of the class

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

In object-oriented programming, a class needs to manually call a superclass's \_\_init\_\_ method in order to ensure that the superclass's initialization logic is executed. This is particularly important when the subclass needs to inherit and initialize attributes or perform additional setup defined in the superclass

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

To augment, or extend, an inherited method without completely replacing it, you can override the method in the subclass and call the superclass's version of the method using the super() function. This allows you to add additional functionality to the inherited method while still utilizing the original behavior defined in the superclass

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

The local scope of a class and a function differ in their behavior and the variables they encompass. Here are the key differences:

Variables: In a class, variables defined within the class body are considered class-level variables or attributes.

Accessibility: Class-level variables in the local scope of a class can be accessed within any method of the class using the self keyword or the class name.

Lifetime: Class-level variables have a lifetime equal to the lifetime of the class instance. They persist as long as the class instance exists