**Python advance assignment-3**

1. **What is the concept of an abstract superclass?**

An abstract superclass is a class that serves as a blueprint for other classes, but cannot be instantiated itself. It defines common properties and methods that subclasses can inherit and potentially override.

1. **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 with the given name and value.

1. **Why does a class need to manually call a superclass's init method?**

When a class is defined, if it has a constructor method named \_\_init\_\_, it will be executed automatically when a new instance of the class is created. However, if the class inherits from a superclass and the superclass has an \_\_init\_\_ method, it will not be called automatically. The subclass's \_\_init\_\_ method must call the superclass's \_\_init\_\_ method explicitly using super().\_\_init\_\_()

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

To augment an inherited method, you can create a new method with the same name in the subclass, and use the super() function to call the parent class's method. Then you can add additional functionality before or after calling the parent class's method.

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

The local scope of a class is different from that of a function in that it includes both instance variables (defined within the class's methods) and class variables (defined at the top level of the class). Variables defined in the local scope of a function are only available within that function and are not shared across instances.