1. What is the concept of an abstract superclass?

***Ans:***

abstract classes are class but you cannot create object from it directly. The purpose of this class is to define how other classes should look like, what kind of method and properties they should have. Th methods and properties defined in abstract class are known as abstract method and abstract properties. All abstract methods and properties need to be implemented in a child class in order to create object from it.

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

***Ans:*** the assignment statement creates a class level variable that is shared among all instances of the class.

Example:

Class car:

Color = “green”

In the above example, the [color = “green”] assignment statement creates a class-level variable called ‘color’ that is shared among all instances of the ‘car’ class. This mean any changes made to ‘color’ through one instance of the class will affect the value of ‘color’ in all other instances of the class.

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

***Ans:***

When a class is derived from a superclass. it inherits all the attributes and methods of the superclass. This includes the \_\_init\_\_() method, which is used to initialize the object's state.

However, if the subclass defines its own \_\_init\_\_() method, it may not automatically call the superclass's \_\_init\_\_() method, which can lead to issues if the superclass has important initialization logic that needs to be executed before the subclass's initialization logic.

To ensure that the superclass's \_\_init\_\_() method is called when a subclass is instantiated, the subclass must explicitly call the superclass's \_\_init\_\_() method. This can be done using the super() function, which returns a temporary object of the superclass and allows you to call its methods.

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

***Ans:***

we 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 your own functionality to the method while still retaining the functionality of the superclass's version of the method.

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

***Ans:***

1. class defines a new namespace and function defines a new local namespace within the current namespace of the class.
2. Variable which are defined within class are accessible throughout the class and its instances, but variable defined within a function are only accessible within the function’s local namespace.
3. In a class the ‘self’ parameter is used to refer to the current instance of the class, while in a function, parameters are used to pass in arguments.
4. Methods defined within a class can access and modify the state of the object while functions generally cannot access or modify the state of the object.