**Python Advanced Assignment 3**

1. What is the concept of an abstract superclass?

Ans-) An abstract superclass is a class that is designed to be inherited by other classes but cannot be instantiated on its own. It defines a common interface and basic behavior that its subclasses can inherit and override as necessary. The purpose of an abstract superclass is to provide a general framework for a group of related classes, while allowing for specific behavior to be implemented in each subclass.

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 creates a class variable with the specified name and assigns it the specified value. This variable is shared by all instances of the class, and can be accessed and modified by any method of the class or its instances.

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

Ans-) A class needs to manually call a superclass's init method if it wants to inherit and customize the behavior of the superclass's constructor. By calling the superclass's init method in the subclass's constructor, the subclass can perform any necessary initialization before or after the superclass's initialization, and can also pass additional arguments to the superclass if needed.

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

Ans-) To augment an inherited method, you can override the method in the subclass and call the superclass's implementation using the super() function. This allows you to add behavior before or after the superclass's behavior, or modify its behavior in some way without completely replacing it.

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

Ans-) The local scope of a class is different from that of a function in several ways. First, a class defines a new namespace that is separate from the global namespace and any other namespaces defined in the program. Second, the scope of a class's methods is determined by the class's namespace, not by the local scope of any individual method. Third, class variables are shared by all instances of the class, while function variables are local to the function and not shared by other functions or instances. Finally, a class can have a hierarchy of nested scopes, with nested classes or functions defining their own local scopes within the class's namespace.